

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-244901

(43)Date of publication of application : 30.08.2002

(51)Int.Cl.

G06F 12/00

G06F 17/30

(21)Application number : 2001-041516

(71)Applicant : SONY CORP

(22)Date of filing : 19.02.2001

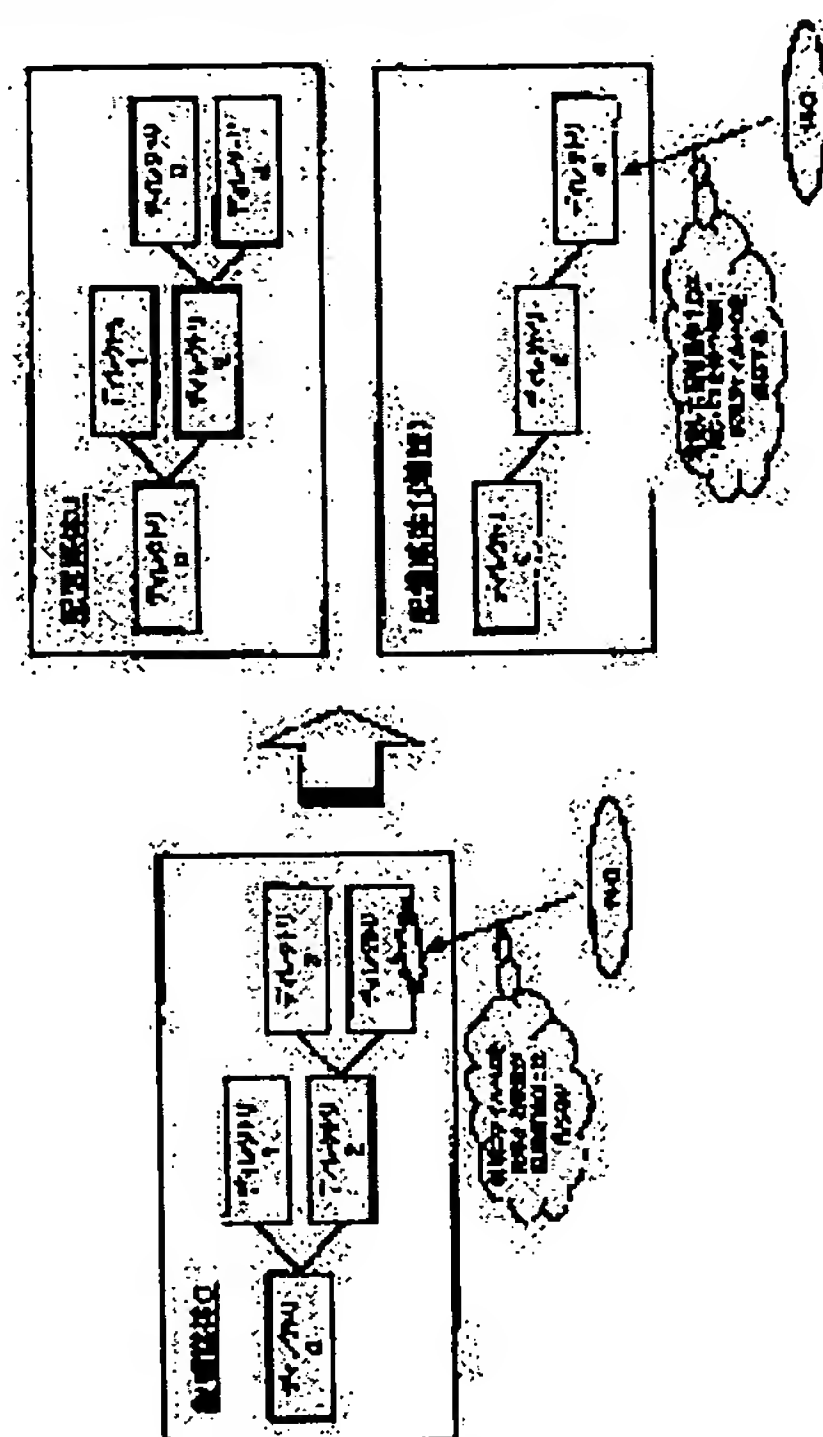
(72)Inventor : NAKABASHI HIDEMICHI

(54) FILE MANAGEMENT DEVICE AND METHOD THEREFOR AND STORAGE MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a file storage function capable of adding a storage medium to increase storage capacity without changing a tree structure of directory seen by a user.

SOLUTION: If storage medium capacity for storing a file is insufficient, the file is stored by making on the storage medium to which the same path as a path to a file storage place on the medium is added. Moreover, if the file cannot be stored in an existing storage medium because an existing file is edited and its size becomes too large, a path up to the directory which has been storage destination to the present time is generated on another medium having vacant capacity to store the file there.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The 1st storage with which it is file management equipment for introducing the tree structure into the storage space which consists of two or more storages, and managing a file hierarchical, and the predetermined tree structure was formed, A pass generation means to generate the same pass for arriving at the predetermined location on the tree structure in the 2nd storage with which the availability was secured, and said 1st storage on said 2nd storage, File management equipment characterized by providing a file preservation means to save a file at said 2nd storage using the generated this pass.

[Claim 2] Said pass generation means is file management equipment according to claim 1 characterized by what it answers that there was no availability which should save a file at said 1st storage, and the same pass for arriving at the location which should arrange this file on said tree structure is generated for on said 2nd storage.

[Claim 3] It has further a file-editing means to edit the file on each storage. Said pass generation means It answers that the availability which should be saved on said 1st storage as a result to which the file on said 1st storage was edited into, and file size increased was lost. File management equipment according to claim 1 characterized by what the same pass for arriving at the location which should arrange this file on said tree structure is generated for on said 2nd storage.

[Claim 4] It is file management equipment according to claim 1 which is further equipped with a tree structure creation means to create the tree structure on each storage, and is characterized by what the same pass for reaching the tree structure which said pass generation means should answer that the availability for creating the tree structure on said 1st storage was lost, and should be this created is generated for on said 2nd storage.

[Claim 5] File management equipment according to claim 1 characterized by what it has further a file search means to search for whether the file of the same file name as the same pathname exists on each storage for.

[Claim 6] It is the file management approach for managing a file hierarchical on the storage space which consists of two or more storages containing the 1st storage with which the predetermined tree structure was formed, and the 2nd storage with which the availability was secured. The pass generation step which generates the same pass for arriving at the predetermined location on the tree structure in said 1st storage on said 2nd storage, The file management approach characterized by providing the file preservation step which saves a file at said 2nd storage using the generated this pass.

[Claim 7] The file management approach according to claim 6 characterized by what it answers that there was no availability which should save a file at said 1st storage at said pass generation step, and the same pass for arriving at the location which should arrange this file on said tree structure is generated for on said 2nd storage.

[Claim 8] It has further the file-editing step which edits the file on each storage. At said pass generation step It answers that the availability which should be saved on said 1st storage as a result to which the file on said 1st storage was edited into, and file size increased was lost. The file management approach according to claim 6 characterized by what the same pass for arriving at the location which should

arrange this file on said tree structure is generated for on said 2nd storage.

[Claim 9] The file management approach according to claim 6 characterized by what the same pass for reaching the tree structure which should be further equipped with the tree structure creation step which creates the tree structure on each storage, should answer that the availability for creating the tree structure on said 1st storage was lost at said pass generation step, and should be this created is generated for on said 2nd storage.

[Claim 10] The file management approach according to claim 6 characterized by what it has further for the file search step which searches for whether the file of the same file name as the same pathname exists on each storage.

[Claim 11] So that processing for managing a file hierarchical on the storage space which consists of two or more storages containing the 1st storage with which the predetermined tree structure was formed, and the 2nd storage with which the availability was secured may be performed on computer system It is the storage which stored the described computer software physically in the computer-readable format. Said computer software The pass generation step which generates the same pass for arriving at the predetermined location on the tree structure in said 1st storage on said 2nd storage, The storage characterized by providing the file preservation step which saves a file at said 2nd storage using the generated this pass.

[Translation done.]

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the file management equipment and the file management approach for managing the file of a large number digitized possible [handling] on computer system, and relates to the file management equipment and the file management approach for introducing the tree structure of a directory etc. into a storage space, and managing a file hierarchical especially.

[0002] Furthermore, it is related with the file management equipment and the file management approach to which start the detailed file management equipment which manages much files using directory structure on the computer by which this invention was constituted possible [addition of two or more storages, such as a hard disk,], and the file management approach, add a storage, without changing the tree structure of the directory especially seen from the user, and storage capacity is made to increase.

[0003]

[Description of the Prior Art] In connection with the technological innovation of these days, the spread of various kinds of information processing and information communication equipment progresses quickly, and is permeating office, ordinary homes, etc. deeply. Various data, such as not only mere text data but an image and voice, are conjointly digitized with this, and it has come to be dealt with on a computer in the format of the so-called "file."

[0004] Generally computer filing is saved under management of the file system in an operating system memory, a hard disk, a floppy (trademark) disk, or on various kinds of archive media in addition to this. Moreover, a file system can manage much files now hierarchical by introducing the tree structure called a "directory."

[0005] In a general file system, the tree structure of the directory which became independent per the media unit or drive, respectively is formed like a hard disk drive, a floppy disk drive, and a CD-ROM drive. Therefore, when the directory structure formed within the existing specific drives, such as a hard disk, is convenient and the same directory structure as other drives or other media is applied, directory structure must be anew created on other drives by the user's himself system operation.

[0006] On the other hand, there is a limitation in the storage capacity equipped standardly with computer system according to a limit of cost or file space, and other design-situations. For this reason, in almost all systems, the memory storage function which can add storages, such as a hard disk, is equipped.

[0007] In order to perform file management suitably in the extended hard disk, directory structure must newly be determined. For example, from the operating system represented by "Windows" of U.S. Microsoft Corp., new directory structure will be made on the extended hard disk, and the file will be stored there with it.

[0008] By such technique, even if a user makes directory structure for a classification and filing of a file and stores the file there, when using a still newer hard disk, another directory in addition to the existing directory on the hard disk which was being used until now will be made serially, and the new file there will be stored. It will be tormented by such troublesomeness, whenever the capacity of a hard disk is

insufficient and it adds a new hard disk.

[0009] By the file management approach left to manual actuation of such a user, the files which belong to the same classification and should be stored in the directory same originally are stored in a different hard disk depending on a file creation stage or a preservation stage, or a different directory, and they become things. That is, the file which should be carried out a group to the same classification as a result by which hard disk duplication, new directory creation, etc. are inserted in between will distribute to two or more directories. In such a case, though he wants to come to refer to a file with relation when perusing or editing one file later, although there is nothing to the same directory therefore, it may be unable to pull out. Furthermore, a storing location cannot be remembered but it is buried into file space, and it may remain missing.

[0010] Moreover, in the operating system of the Unix system widely known in this industry as the object for development environment, or an object for servers, there is no change in a new directory still generating one directory structure by mounting (Mount: enabling it to access peripheral devices, such as a hard disk, on a computer), although the file manipulation nature on nothing and a desktop is raised. A symbolic link can arrange. However, there is no change in that the activity whose user stretches an explicit link is the need, and being complicated for a user, if it takes into consideration that directory structure actually (physical) changes.

[0011] Although the file memory space of infinity can be mostly offered by duplication of a hard disk etc. in the present computer system, this cannot but be the effectiveness that the storage capacity of a system only increases. Namely, if it sees in the position of the user who operates it on a desktop screen, the added storage will only increase the memory capacity of a system, and will not contribute it to the file classification by the directory. If a disk is extended, it is necessary to make the tree structure incidental to the disk independently with former. The directory created on the extended disk coexists with the tree structure before duplication, and classification of a file becomes complicated.

[0012]

[Problem(s) to be Solved by the Invention] The object of this invention is to offer the outstanding file management equipment and the outstanding file management approach of being able to introduce the tree structure of a directory etc. into a storage space, and managing a file hierarchical.

[0013] The further object of this invention is to offer the outstanding file management equipment and the outstanding file management approach much files are manageable on the computer constituted possible [addition of storages, such as a hard disk,] using directory structure.

[0014] The further object of this invention is to offer the outstanding file management equipment and the outstanding file management approach to which a storage can be added, without changing the tree structure of the directory seen from the user, and storage capacity can be made to increase.

[0015]

[Means for Solving the Problem and its Function] This invention is made in consideration of the above-mentioned technical problem. The 1st side face The 1st storage with which it is file management equipment for introducing the tree structure into the storage space which consists of two or more storages, and managing a file hierarchical, and the predetermined tree structure was formed, A pass generation means to generate the same pass for arriving at the predetermined location on the tree structure in the 2nd storage with which the availability was secured, and said 1st storage on said 2nd storage, It is file management equipment characterized by providing a file preservation means to save a file at said 2nd storage using the generated this pass.

[0016] Moreover, the 2nd side face of this invention is the file management approach for managing a file hierarchical on the storage space which consists of two or more storages containing the 1st storage with which the predetermined tree structure was formed, and the 2nd storage with which the availability was secured. The pass generation step which generates the same pass for arriving at the predetermined location on the tree structure in said 1st storage on said 2nd storage, It is the file management approach characterized by providing the file preservation step which saves a file at said 2nd storage using the generated this pass.

[0017] Said pass generation means or step can answer that there was no availability which should save a

file at said 1st storage, and can generate the same pass for arriving at the location which should arrange this file on said tree structure on said 2nd storage.

[0018] Therefore, when according to the file management equipment or the approach concerning the 1st or 2nd side face of this invention it is going to store a certain file and the capacity of a storage is lacking, it can make on the storage which extended the same pass as the pass to the file storing location on the medium, and the file concerned can be stored there.

[0019] That is, according to the 1st or 2nd side face of this invention, even if it is after extending a storage, for example, a new file is stored in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user should just perform drag-and-drop actuation of the mouse of placing a file into the corresponding directory, file manipulation nature is brief.

[0020] Moreover, the file management equipment or the approach concerning the 1st or 2nd side face of this invention may be further equipped with the file-editing means or step which edits the file on each storage. In such a case, said pass generation means or step answers that the availability which should be saved on said 1st storage as a result to which the file on said 1st storage was edited into, and file size increased was lost, and should just generate the same pass for arriving at the location which should arrange this file on said tree structure on said 2nd storage.

[0021] That is, according to the 1st or 2nd side face of this invention, the file to which size became large by the editing task is stored on other storages in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI environment was offered, since a user should just perform drag-and-drop actuation of the mouse of placing a file into the corresponding directory, file manipulation nature is brief.

[0022] Moreover, the file management equipment or the approach concerning the 1st or 2nd side face of this invention may be further equipped with the tree structure creation means or step which creates the tree structure on each storage. In such a case, said pass generation means or step should just generate the same pass for reaching on said 1st storage at the tree structure which should answer that the availability for creating the tree structure was lost, and should be this created on said 2nd storage.

[0023] That is, according to the 1st or 2nd side face of this invention, a new directory is generated on another storage in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI environment was offered, since a user can access a new directory according to the usual mouse actuation, without being conscious of distinction of each storage, file manipulation nature is brief.

[0024] Moreover, the file management equipment or the approach concerning the 1st or 2nd side face of this invention may be further equipped with the file search means or step which searches for whether the file of the same file name as the same pathname exists on each storage.

[0025] According to this file search means or step, even if it extends a storage, any files can be read in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI environment was offered, since a user can access a file according to the usual mouse actuation, without being conscious of whether the desired file is arranged at which storage, file manipulation nature is brief.

[0026] A deer is carried out, and if the file management technique concerning this invention is followed, from a user, it seems that unific directory structure is formed on the single storage region which a system offers. therefore -- even if the storage was added, from the user, the capacity of a storage region increased simply -- **** -- it is not visible and file manipulation can be performed only being conscious of old directory structure.

[0027] Each directory is physically distributed on a different storage, and it is arranged by the repeat of duplication of a storage. According to this invention, a file is stored also to a directory in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI environment was offered, since a user only performs drag-and-drop actuation of a mouse and can perform the usual file manipulation of being as delete **** [, and], file manipulation nature is brief.

[saving a file to a desired directory] [read] [move]

[0028] Moreover, the 3rd side face of this invention So that processing for managing a file hierarchical on the storage space which consists of two or more storages containing the 1st storage with which the predetermined tree structure was formed, and the 2nd storage with which the availability was secured may be performed on computer system It is the storage which stored the described computer software physically in the computer-readable format. Said computer software The pass generation step which generates the same pass for arriving at the predetermined location on the tree structure in said 1st storage on said 2nd storage, It is the storage characterized by providing the file preservation step which saves a file at said 2nd storage using the generated this pass.

[0029] The storage concerning the 3rd side face of this invention is a medium which offers computer software in a computer-readable format to the general purpose computer system which can perform various program codes, for example. Attachment and detachment of CD (Compact Disc), FD (Floppy Disk), MO (Magneto-Optical disc), etc., etc. are free for such a medium, and it is a storage of portability. Or it is also technically possible to provide specific computer system with computer software via transmission media, such as a network (for a network not to ask distinction of wireless and a cable), etc.

[0030] Such a storage defines the collaboration-relation on the structure of the computer software and the storage for realizing the function of computer software predetermined in a computer system top, or a function. If it puts in another way, by installing predetermined computer software in computer system through the storage concerning the 3rd side face of this invention, on computer system, a collaboration-operation is demonstrated and the same operation effectiveness as the file management equipment and the approach concerning each 1st [of this invention] and 2nd side faces can be acquired.

[0031] The object, the description, and advantage of further others of this invention will become [rather than] clear by detailed explanation based on the example and the drawing to attach of this invention mentioned later.

[0032]

[Embodiment of the Invention] The file management equipment or the file management approach concerning this invention adds a storage, without changing the tree structure of the directory seen from the user, makes memory capacity increase on the computer constituted possible [addition of storages, such as a hard disk,], and makes brief the file manipulation and the file classification by the user on a desktop. Here, the procedure of the file management concerning this invention is explained roughly first.

[0033] (1) In order to classify and carry out file management on storages, such as a hard disk, at the time of new file preservation, generally, create directory structure and save each file in the corresponding delay chestnut.

[0034] A new storage is extended, when it is going to store a certain file and the capacity of a storage is lacking. At this time, it creates also on the storage to which the same pass as the pass to the storing location of the file on the original storage was added, and the file concerned is created there. Even if the procedure adds how many storages, it is the same.

[0035] The procedure of the file management performed when extending a storage in drawing 1 and saving a new file at it is shown typically.

[0036] In the example shown in this drawing, first, while the system is equipped with the storage 0, subdirectories 1 and 2 are formed in zero or less root directory in this storage 0, and those subdirectories 3 and 4 are further formed in two or less directory.

[0037] Here, when you try to save the new file f40 in a directory 4, suppose that the availability of a storage 0 was insufficient and it was not able to save.

[0038] In such a case, since the new file f40 is saved, the new storage 1 is extended to a system.

[0039] According to this invention, while arranging a subdirectory 2 on a storage 1 at zero or less root directory, a subdirectory 4 is arranged to two or less directory, and the same pass as the directory 4 on the original storage 0 is prepared on the new storage 1. And originally the file f40 which should have been saved to the directory 4 on a storage 0 is saved in the directory 4 by the side of the new storage 1.

[0040] That is, according to this invention, after extending a storage 1, the new file f40 is stored by

internal processing of a system in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user should just perform drag-and-drop actuation of the mouse of placing a file f40 into a directory 4, file manipulation nature is brief.

[0041] (2) By editing the file which already exists at the time of file editing, the file size may become larger than before, and the availability to store may not remain on the original storage.

[0042] In such a case, what is necessary is just to save there, if the pass of the directory which stores the file concerned on another storage as for which capacity is vacant is already prepared. Moreover, if pass is not yet prepared, the pass to the directory is generated and the file concerned is saved there. If the capacity which generates the directory path is not vacant any longer on another storage, according to the same procedure, it processes on still more nearly another storage.

[0043] Size increases in drawing 2 by an editing task etc., and the procedure of the file management performed when saving the file it became impossible to save at the original storage on another storage is typically shown in it.

[0044] In the example shown in this drawing, the storage 0 equipped from origin and the storage 1 by which additional duplication was carried out later exist in a system. Moreover, subdirectories 1 and 2 are formed in zero or less root directory in a storage 0, and the subdirectories 3 and 4 are further formed in two or less directory. On the storage 1 of another side, while arranging a directory 2 to the zero or less same directory path as the storage 0 for referring to the file in a directory 4, i.e., a directory, the directory path of arranging a directory 4 is formed in two or less directory.

[0045] Here, if the file f30 in a directory 3 is edited on a storage 0, it will carry out to the availability of a storage 0 having been insufficient and it having become impossible to save.

[0046] In such a case, the pass to the directory 3 for accessing the file f30 after edit distinguishes first whether it exists by ** on the extended storage 1. If such a directory path exists, a file f30 is saved on a storage 1 using it.

[0047] In the example shown in drawing 2, the pass to the directory 3 for accessing a file f30 is not prepared yet on the storage 1. Then, also on a storage 1, by arranging a directory 3 to two or less subdirectory of a root directory 0, the same directory path as a storage 0 is generated, and a file f30 is saved in the directory 3.

[0048] Finally the file f30 of the origin which was in the directory 3 of a storage 0 is eliminated, and contention with the file f30 newly saved on the storage 1 is avoided. Elimination of the original file leads also to release of the storage space of a storage 0.

[0049] That is, according to this invention, after extending a storage 1, the file f30 to which size became large is stored by internal processing of a system in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user should just perform drag-and-drop actuation of the mouse of placing a file f30 into a directory 3, file manipulation nature is brief.

[0050] In addition, although the pass to the directory 3 for accessing a file f30 on a storage 1 although not illustrated is already prepared, when there is no availability sufficient also on a storage 1, generation of the same directory path as the above and the activity of file preservation are done on another storage with sufficient availability.

[0051] (3) new directory creation time -- generate the pass to the directory on the storage as for which sufficient memory capacity to generate the directory is vacant to create a new directory.

[0052] Consequently, since it has the same directory path even if a directory is arranged on which storage, a user can access the file in a directory, without being conscious of the physical location of a directory.

[0053] The procedure of the file management performed when creating the directory it became impossible to create to the original storage on another storage is typically shown in drawing 3.

[0054] In the example shown in this drawing, the storage 0 equipped from origin and the storage 1 by which additional duplication was carried out later exist in a system. Moreover, subdirectories 1 and 2 are formed in zero or less root directory in a storage 0, and the subdirectories 3 and 4 are further formed in

two or less directory. On the storage 1 of another side, while arranging a directory 2 to the zero or less same directory path as the storage 0 for referring to the file in a directory 4, i.e., a directory, the directory path of arranging a directory 4 is formed in two or less directory.

[0055] Here, although it is going to generate the new directory 5 to two or less directory on a storage 0, suppose that the memory capacity of a storage 0 was insufficient and a directory was not able to be generated.

[0056] In such a case, in order to generate a new directory on the extended storage 1 first, it distinguishes whether there is any sufficient availability. And a directory 5 will be generated if there is sufficient availability, two or less location, i.e., the directory, which has the same pass on a storage 1.

[0057] That is, according to this invention, after extending a storage 1, the new directory 5 is generated by internal processing of a system in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user can access the new directory 5 according to the usual mouse actuation, without being conscious of whether it shall be arranged between the storage 0 or the storage 1 at any, file manipulation nature is brief.

[0058] In addition, although not illustrated, when there is no availability sufficient also on a storage 1, a new directory is generated in the location which has the same pass as the above on another storage with sufficient availability.

[0059] (4) When reading a file at the time of file read-out, look for each storage in order based on a pathname and a file name.

[0060] When generating a new file to the directory of a certain pass, it looks for a storage in order and is made whether there is any file of the same pathname and a file name, and for problems, such as overwrite of a file, not to appear.

[0061] The sequence of looking for a storage may make a head the storage equipped standardly, for example, and may be the extended sequence hereafter.

[0062] The procedure of the file management performed when carrying out file reference ranging over two or more storages is typically shown in drawing 4.

[0063] In the example shown in this drawing, the storage 0 equipped from origin, and the storage 1 and storage 2 by which additional duplication was carried out later exist in a system. Subdirectories 1 and 2 are formed in zero or less root directory in a storage 0, and the subdirectories 3 and 4 are further formed in two or less directory. Moreover, on the storage 1, while arranging a directory 2 to the zero or less same directory path as the storage 0 for referring to the file in a directory 4, i.e., a directory, the directory path of arranging a directory 4 is formed in two or less directory. Moreover, on the storage 2, while arranging a directory 2 to the zero or less same directory path as the storage 0 for referring to the file in a directory 5, i.e., a directory, the directory path of arranging a directory 5 is formed in two or less directory.

[0064] Here, suppose that the command of the file f51 saved in the zero or less-root directory subdirectory [two or less-subdirectory] 5, i.e., the purport which reads "Dir0/Dir2 / Dir5/f51", was published on the file space of a system.

[0065] In such a case, tree retrieval is performed [whether a directory 5 exists in two or less zero or less root directory / on a storage 0 / subdirectory, and] first. In the example shown in this drawing, a directory 5 does not exist on a storage 0. Then, tree retrieval is performed [whether a directory 5 exists in two or less zero or less-root directory subdirectory, and] also on the extended storage 1. In the example shown in this drawing, a directory 5 does not exist on a storage 1. Then, tree retrieval is further performed [whether a directory 5 exists in two or less zero or less-root directory subdirectory, and] also on the extended storage 2. And since a directory 5 can be discovered on the success reverse side on a storage 2, it reads in the directory and read-out access is carried out at the target file f51.

[0066] That is, according to this invention, after extending a storage 1, any files can be read by internal processing of a system in the condition which can be referred to with the same pass as the former. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user can access a file according to the usual mouse actuation, without being conscious of whether

the desired file shall be arranged between the storage 0 or the extended storage 1, and the storage 2 at any, file manipulation nature is brief.

[0067] When file management is performed according to this invention, the directory structure reflected in a user is typically shown in drawing 5.

[0068] The storage with which plurality became independent actually in the system lives together, and each directory is physically distributed and arranged ranging over two or more storages. However, if the file management technique concerning this invention mentioned above is followed, from a user, it seems that unific directory structure is formed on the single storage region which a system offers.

[0069] If it says in the example shown in drawing 5, from the user, a subdirectory 1 and a subdirectory 2 are arranged at zero or less root directory, and the subdirectory 5 is arranged further in the subdirectory 3, the subdirectory 4, and the list at two or less subdirectory.

[0070] Although dispersedly arranged on the storage from which each directory differs physically by the repeat of duplication of a storage, according to this invention, a file is stored to any directories in the condition which can be referred to with the same pass as the former by internal processing of a system. On the desktop screen where for example, the GUI (Graphical User Interface) environment was offered, since a user only performs drag-and-drop actuation of a mouse and can perform the usual file manipulation of being as delete **** [, and], file manipulation nature is brief. [saving a file to a desired directory] [read] [move]

[0071] When the above is summarized, according to the file management approach, from a user, it seems that there is a storage region irrespective of how many storages exist on a system in the file management equipment list concerning this invention. therefore -- even if the storage was added, from the user, the capacity of a storage region increased simply -- **** -- it is not visible and file manipulation can be performed only being conscious of old directory structure.

[0072] Subsequently, the operation gestalt of this invention is explained concretely.

[0073] Although file management equipment equipped with the hardware structure of dedication can also be designed in order to realize this invention, it is the gestalt of performing a program code predetermined in the computer system top of the versatility which can perform application of varieties, and it is also possible to mount this invention. The hardware configuration of the computer system 100 which can realize file management concerning this invention is typically shown in drawing 6. Hereafter, each block is explained according to this drawing.

[0074] CPU (Central Processing Unit)101 which is the Maine controller of a system 100 performs various kinds of applications under control of an operating system (OS). CPU101 interconnects with other equipments (after-mentioned) by bus 108 as the graphic display.

[0075] Memory 102 is storage used since the program code performed in CPU101 is stored or the activity data under activation are stored temporarily. Moreover, memory 102 can function also as a storage for file management or file preservation.

[0076] The display controller 103 is an exclusive controller for processing actually the drawing instruction which CPU101 publishes. Once it is written in a frame buffer (not shown), the screen output of the drawing data processed in the display controller 103 is carried out on a display 111.

[0077] The user input interface 104 is equipment for connecting user input devices, such as a keyboard (KBD) 112 and a mouse 113, to computer system 100.

[0078] For example, the user input environment of the interactive graphic base and GUI (Graphical User Interface) ***** are offered with the combination of the "desktop" offered on the screen of a display 111, and these user input equipment 112,113. In such a case, the user of a system 100 can input the user command to a corresponding file and application intuitively by impressing actuation, such as a click and drag and drop, through a mouse 113 on a desktop to the icon (pictorial symbol) assigned to each file and application. Of course, preservation of the file to a directory, file migration between file read-out and a directory, etc. are performed considering mouse actuation as the keynote.

[0079] The network interface 105 can connect a system 100 to local networks, such as LAN (Local Area Network), and a wide area network still like the Internet according to predetermined communications protocols, such as Ethernet.

[0080] On the network, two or more host terminals, such as a contents receiving set, are connected in the transparent condition, and the distributed computing environment is built. For example, the predetermined file from a server and various contents on a network are downloadable. Of course, the computer software for realizing file management concerning this invention is also downloadable via a network.

[0081] The external instrument interface 107 is equipment for connecting two or more storages 150A and 150B and -- to this computer system 100. Based on interface specifications, such as IDE (Integrated Drive Electronics) and SCSI (Small Computer System Interface), the design and the fabrication of the external instrument interface 107 are done.

[0082] Each storage 150A and 150B -- Upwards, various kinds of data and contents by file format are storable. Storages 150A and 150B -- Computer system 100 is equipped standardly at least with a part inside, and others are extended. The maximum number of the storage which can be extended is determined according to the interface specification adopted as the external instrument interface 107. Storage 150A and 150B-- consist of storages which can be written, such as a hard disk drive, MO (Magneto-Optical disc) drive, and a DVD(Digital Versatile)-RAM drive. Of course, the read-only storage may be included like a CD-ROM drive or a DVD-ROM drive in the part.

[0083] A hard disk drive is the external storage which carried the magnetic disk as storage support fixed (common knowledge), and excels other external storage in points, such as memory capacity and a data transfer rate. It calls it "install" to the system of a program to place on a hard disk in the condition that a software program can be performed. Usually, the program code which CPU101 should perform is stored in the hard disk in un-volatilizing. For example, the software which performs file management concerning the operation gestalt of this invention is installable on a hard disk. Of course, the multiple files put under this file management can also be saved on a hard disk.

[0084] When a storage consists of exchangeable portable mold media, such as CD (Compact Disc), and MO (Magneto-Optical disc), DVD (Digital Versatile Disc), it is used in order to move backing up a software program, a data file, etc. as data of a computer-readable format, and these between systems (that is, a sale, a negotiation, and distribution are included). For example, these portable mold media can be used, and the computer software which performs file management concerning the operation gestalt of this invention can be physically circulated and distributed among two or more devices. Of course, portable mold media can be used, and the multiple files put under this file management can also be physically circulated and distributed between devices.

[0085] In addition, an example of the computer system 100 as shown in drawing 6 is the compatible machine or succeeding machine of personal computer"PC/AT(Personal Computer/Advanced Technology)" of U.S. IBM. Of course, it is also possible to apply the computer equipped with other architecture as computer system 100 concerning this operation gestalt.

[0086] Subsequently, the file management approach realized on the computer system 100 mentioned above is explained.

[0087] 1. In operation gestalt 1 computer system 100, generally, the device driver is prepared for every peripheral device, and high order software, such as an operating system, generalizes the whole system by controlling device actuation of corresponding through a device driver. Of course, the device driver is arranged at every each storage 150A mentioned above and 150B--.

[0088] With this operation gestalt, one storage 150A with which computer system 100 was equipped has special semantics, and the object of writing in LUT (Look Up Table) which manages the storing information on the file in a system 100 and the information on the tree structure is given to this. The tree structure currently written to this LUT is the tree structure (in this description, it considers as "a tree 0" below) which exists only in one system. Although each storage 150A and 150B-- have the tree structure, respectively, please grasp that each tree structure is the system-wide tree structure, i.e., the subset of a tree 0.

[0089] The relation between the tree structure which each storage 150A and 150B-- have, and the tree structure "a tree 0" of the whole system is as follows. namely, the time of storing a certain file F0 in a directory called /do/d1/d2/d3 -- a user -- not being visible (or not visualized on the desktop) -- suppose

that the system tends to store in the Kth storage. If a directory called /d0/d1/d2/d3 does not exist in the Kth storage at this time, a directory called /d0/d1/d2/d3 will be first created to this Kth storage. And a file F0 is stored there. As a result of making internal control of such a system 100, the original tree structure as partial collating of the tree 0 which is the system-wide tree structure is built by each storage.

[0090] With the 1st operation gestalt of this invention, the file storing information on a system 100 is written to LUT. As information currently written, they are the information showing the tree structure, the file name stored in each directory, the number (or identification information) of the storage with which the file is stored actually, the number of the storage equipped in the system 100, etc. LUT is used when displaying directory structure and a file list on a desktop on the bottom of the GUI environment. Moreover, LUT is used as the information source of the retrieval at the time of file read-out. Moreover, when storing of a file, migration, copy, deletion, etc. are operated, the information on LUT is rewritten serially.

[0091] Below, the concrete procedure of the file management in the 1st operation gestalt of this invention is explained.

[0092] (1) It is going to store the new file F0 in a certain directory, /d0 [for example,],/d1/d2 on the view of the storing user of a new file, or a desktop. If there is sufficient opening for a total storing capacity of a system 100, a file F0 is stored as it is in d/d0/d 1-/2. On the other hand, if there is not sufficient opening for a total storing capacity of a system 100, it will be in an error situation and a user will be notified of the message of the purport which a file F0 cannot store (for example, a dialog appears on a desktop).

[0093] The motion inside the system for storing a new file is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{ }."

[0094]

[Equation 1]

Number K=1LOOPof storage for storing of N= system 1: IF (/d0/d1/d2 exist in the tree structure of the Kth storage) {-- IF (there is an availability which stores F0 in the Kth storage) {-- Store F0 in /d0 of the Kth storage/d1/d2. Write the information about /d0/d1/d2/F0 in LUT. GOTO END } ELSE {-- IF (K=N) {GOTO ERROR} ELSE {K=K+1 GOTO LOOP1} } } ELSE {-- IF (there is an availability which makes a directory called /d0/d1/d2 to the Kth medium, and stores F0 in it) {-- Make /d0/d1/d2 to the Kth medium. Store F0 in /d0 of the Kth medium/d1/d2. Write the information about /d0/d1/d2/F0 in LUT. GOTO END } ELSE { IF (K=N) {GOTO ERROR} ELSE {K=K+1 GOTOLOOP1} } }

ENDOfLOOP1:ERROR : It is the message of a purport unstorable any longer Issuance END: [0095] (2)

The file F1 which exists in a directory called /d3/d4 tends to be edited on edit / preservation user's view, or a desktop, and it is going to save the existing file by the same pathname and the same directory name. If there is sufficient opening for a total storing capacity of a system 100 at this time, a file F1 will be overwritten as /d3/d4/F1. On the other hand, if there is not sufficient opening for a total storing capacity of a system 100, it will be in an error situation and a user will be notified of the message of the purport which a file F1 cannot store (for example, a dialog appears on a desktop).

[0096] The motion inside the system for saving, after editing an existing file is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{ }."

[0097]

[Equation 2]

The number of the storage with which the file F1 was stored before number M= edit of the storage for storing of N= system

after edit of F1, and IF (the capacity for overwriting /d3/d 4-/F1/is vacant enough in the Mth storage) {-- Store [d4/] /d3 of the Mth storage/F1. Write the information about /d3-/d4/F1/in LUT. GOTO END} ELSE {-- K= 1 LOOP2: IF (K==M&&K==N) {GOTO ERROR} ELSE IF (K==M&&K!=N) {K=K+1;GOTO LOOP2} ELSE {-- IF (/d3/d4 exist in the tree structure of the Kth storage) {-- IF (there is an availability which stores F1 in the Kth medium) {-- Write the information about /d3/d4/F1 in LUT which stores F1 in /d3/d4 of the Kth medium. GOTO END } ELSE {-- IF (K==N) {GOTO ERROR} ELSE {K=K+1 GOTO LOOP2} } }ELSE {IF (there is an availability which makes a directory called /d3/d4 to the Kth medium, and stores F1 in it) Make /d3/d4 of a { K position medium. Store [d4/] /d3 of the Kth medium/F1. GOTO END which writes the information about /d3/d4/F1 in LUT } ELSE { IF (K==N) {GOTO ERROR} ELSE {K=K+1 GOTO LOOP1} } } } ENDOfLOOP2:}ERROR: It is the message of a not storable purport Issuance END: [0098] (3) Impress the actuation which reads the file F2 which exists in a directory called /d5/d6/d7/d8 on the view of the read-out user of a file, or a desktop. Within the storage space of a system 100, if /d5/d6/d7/d8/F2 exist, it will be read. If there is nothing, it will be in an error situation, and a user is notified of the message that F2 does not exist in the pass (for example, a dialog appears on a desktop).

[0099] The motion inside the system for reading a file from the storage space of a system 100 is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{}."

[0100]

[Equation 3]

IF which refers to LUT (the entry about /d5/d6/d7/d8/F2 is in the entry of LUT) {-- It gets to know the number of the storage stored actually by refer to the LUT. (Suppose that it was the Mth medium) /d5/d6/d7/d8/F2 are read from a M position medium GOTO END}ELSE {-- GOTO ERROR} ERROR : END: [0101] which publishes the message of the purport in which a file does not exist (4)

Impress the actuation which copies the file F2 which exists in a directory called /d5/d6/d7/d8 to a directory called /d9 on the view of the copy user of a file, or a desktop. If there is an opening for storing new new file / d9/F2 in the storage capacity of a system 100, a duplicate called /d9/F2 is stored. If there is no availability, a user will be notified of the message of the purport which will be in an error situation and does not have an availability (for example, a dialog appears on a desktop).

[0102] Inside a system 100, /d5/d6/d7/d8/F2 are read first. The procedure at that time is as the above-mentioned "read-out of a (3) File" having explained.

[0103] If it succeeds in this file read-out, subsequently a new file called /d9/F2 will be stored. The procedure is as the above-mentioned "storing of (1) new file" having explained.

[0104] (5) It is going to delete the file called /d10/d11/F3 on the view of the deletion user of a file, or a desktop. It will delete, if the file exists in the storage space of a system 100. If there is nothing, such a file will return the message of the purport not existing to a user (for example, a dialog appears on a desktop).

[0105] The motion inside the system for deleting a file from the storage space of a system 100 is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{}."

[0106]

[Equation 4]

IF which refers to LUT (the entry of /d10/d11/F3 existed) {-- Delete the entry about /d10 in LUT/d11/F3. Publish the deletion instruction of /d10/d11/F3 to the driver of the storage of a number with which /d10/d11/F3 are stored. GOTO END}ELSE {-- GOTO ERROR}ERROR : the message that there is no file which should be deleted -- issuance END: [0107] (6) Impress the actuation which moves a file called /d5/d6/d7/d8/F2 to other directories, for example, /d12-/d13/, on the view of the migration

user of a file, or a desktop. It will become an error if a capacity required as workspace for migration is not vacant enough to a system 100.

[0108] Inside a system 100, /d5/d6/d7/d8/F2 are first copied as /d12/d13/F2. The procedure at that time is as the above-mentioned "(4) Copy of a file" having explained. Then, if an error occurs, a series of actuation will be stopped.

[0109] If it is not an error, /d5/d6/d7/d8/F2 will be deleted. The procedure at that time is as the above-mentioned "(5) Deletion of a file" having explained.

[0110] 2. Assume that one software which controls in generalization the storage which exist in a system 100 exists with a two operation gestalten operation gestalt. One storage has special semantics and the object of writing in LUT (Look Up Table) which manages the storing information on the file in a system 100 and the information on the tree structure is given to this.

[0111] With the 2nd operation gestalt of this invention, the file storing information on a system 100 is written to LUT. It is the number of the storage equipped in which field is vacant as information currently written in the physical address of the number of the information showing the tree structure, the file name stored in each directory, and the storage with which the file is stored actually, and a file within the storage, and each storage, availability information, and a system 100 etc.

[0112] Below, the concrete procedure of the file management in the 2nd operation gestalt of this invention is explained.

[0113] (1) It is going to store the new file F0 in a certain directory, /d0 [for example,],/d1/d2 on the view of the storing user of a new file, or a desktop. If there is sufficient opening for a total storing capacity of a system 100, a file F0 is stored as it is in d/d0/d 1-/2. On the other hand, if there is not sufficient opening for a total storing capacity of a system 100, it will be in an error situation and a user will be notified of the message of the purport which a file F0 cannot store (for example, a dialog appears on a desktop).

[0114] The motion inside the system for storing a new file is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{}."

[0115]

[Equation 5]

Number K=1LOOPof storage for storing of N= system 1: IF (there is an availability which stores F0 in the Kth medium) {-- Store F0 in the Kth medium. Register the entry about /d0/d1/d2/F0 into LUT.

GOTO END } ELSE {-- IF (K==N) {GOTO ERROR} ELSE {K=K+1 GOTO LOOP1} }

ENDOfLOOP1:ERROR : It is the message of a purport unstorable any longer Issuance END: [0116] In addition, the information registered into LUT in the pseudo code which is the above is at the number of the medium memorized actually, the address with which it was stored in the double pair, and occupancy capacity.

[0117] (2) The file F1 which exists in a directory called /d3/d4 tends to be edited on edit / preservation user's view, or a desktop, and it is going to save the existing file by the same pathname and the same directory name. If there is sufficient opening for a total storing capacity of a system 100 at this time, a file F1 will be overwritten as /d3/d4/F1. On the other hand, if there is not sufficient opening for a total storing capacity of a system 100, it will be in an error situation and a user will be notified of the message of the purport which a file F1 cannot store (for example, a dialog appears on a desktop).

[0118] The motion inside the system for saving, after editing an existing file is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{}."

[0119]

[Equation 6]

The number of the storage with which the file F1 was stored before number M= edit of the storage for storing of N= system

after edit of F1, and IF (the capacity for overwriting /d3/d 4-/F1/is vacant enough in the Mth storage) {-- Store [d4/] /d3 of the Mth storage/F1. Write the information about /d3-/d4/F1/in LUT. GOTO END} ELSE {-- K= 1 LOOP2: IF (K==M&&K==N) {GOTO ERROR} ELSE IF (K==M&&K!=N) {K=K+1;GOTO LOOP2} ELSE {-- IF (there is an availability which stores F1 in the Kth medium) {-- Store F1 in the Kth medium. Rewrite information, such as a physical address and a segment, about /d3/d4/F1 to LUT. GOTO END } ELSE {-- IF (K==N) {GOTO ERROR} ELSE {K=K+1 GOTO LOOP2} } } ENDOfLOOP2;}ERROR: It is the message of a not storable purport Issuance END: [0120]

(3) Impress the actuation which reads the file F2 which exists in a directory called /d5/d6/d7/d8 on the view of the read-out user of a file, or a desktop. Within the storage space of a system 100, if /d5/d6/d7/d8/F2 exist, it will be read. If there is nothing, it will be in an error situation, and a user is notified of the message that F2 does not exist in the pass (for example, a dialog appears on a desktop). [0121] The motion inside the system for reading a file from the storage space of a system 100 is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{ }."

[0122]

[Equation 7]

IF which refers to LUT (the entry about /d5/d6/d7/d8/F2 is in the entry of LUT) {-- It gets to know the number of the storage stored actually by refer to the LUT. (Suppose that it was the Mth medium) /d5/d6/d7/d8/F2 are read from a M position medium GOTO END}ELSE {-- GOTO ERROR} ERROR : END: [0123] which publishes the message of the purport in which a file does not exist (4) Impress the actuation which copies the file F2 which exists in a directory called /d5/d6/d7/d8 to a directory called /d9 on the view of the copy user of a file, or a desktop. If there is an opening for storing new new file / d9/F2 in the storage capacity of a system 100, a duplicate called /d9/F2 is stored. If there is no availability, a user will be notified of the message of the purport which will be in an error situation and does not have an availability (for example, a dialog appears on a desktop).

[0124] Inside a system 100, /d5/d6/d7/d8/F2 are read first. The procedure at that time is as the above-mentioned "read-out of a (3) File" having explained.

[0125] If it succeeds in this file read-out, subsequently a new file called /d9/F2 will be stored. The procedure is as the above-mentioned "storing of (1) new file" having explained.

[0126] (5) It is going to delete the file called /d10/d11/F3 on the view of the deletion user of a file, or a desktop. It will delete, if the file exists in the storage space of a system 100. If there is nothing, such a file will return the message of the purport not existing to a user (for example, a dialog appears on a desktop).

[0127] The motion inside the system for deleting a file from the storage space of a system 100 is shown below in a pseudo code format. However, unless ("*" slips out of the part pinched by "LOOP*" and "ENDOfLOOP*" by figure) and the "goto" sentence, it is the loop-formation processing by which repeat activation is carried out. Moreover, the sentence which starts in "IF" is a conditional-judgment sentence, and if it is filled, it will perform actuation according to the content of description surrounded by "{ }."

[0128]

[Equation 8]

the message that there is no file which refers to LUT and which should be carried out IF (entry of /d10/d11/F3 existed) deletion -- issuance END: { GOTO END which updates the availability information about the storage with which /d10 in LUT which deletes the entry about /d10 in LUT/d11/F3/d11/F3 are stored etc.} -- ELSE -- { GOTO ERROR} ERROR: [0129] (6) Impress the actuation which moves a file called /d5/d6/d7/d8/F2 to other directories, for example, /d12-/d13/, on the view of the migration user of a file, or a desktop. It will become an error if a capacity required as workspace for migration is not vacant enough to a system 100.

[0130] First, with reference to LUT, the system 100 interior is searched for the entry about /d5/d6/d7/d8/F2, and, inside, this is deleted.

[0131] On LUT, an entry called /d12/d13/F2 is newly made, and the information on a storage number, a physical address, etc. which had been written about /d5/d6/d7/d8/F2 is written in about /d12/d13/F2.

[0132] It has explained in detail about this invention, referring to a specific example more than [addenda]. However, it is obvious that this contractor can accomplish correction and substitution of this example in the range which does not deviate from the summary of this invention.

[0133] On these descriptions, although this invention has been explained based on the example of application over a computer, the summary of this invention is not limited to this. For example, in the device of other types with the file storing function which used the storage of a hard disk, memory, etc. which can be written, the effectiveness of compaction of a file classification can be brought about similarly.

[0134] In short, with the gestalt of instantiation, this invention has been indicated and it should not be interpreted restrictively. In order to judge the summary of this invention, the column of the claim indicated at the beginning should be taken into consideration.

[0135]

[Effect of the Invention] As a full account was given above, according to this invention, the outstanding file management equipment and the outstanding file management approach much files are suitably manageable on the computer constituted possible [addition of storages, such as a hard disk,] using directory structure can be offered.

[0136] Moreover, according to this invention, the outstanding file management equipment and the outstanding file management approach to which a storage can be added, without changing the tree structure of the directory seen from the user, and storage capacity can be made to increase can be offered.

[0137] According to the file management equipment and the file management approach concerning this invention, the file classification in the file storing function which uses storages which can be written, such as a hard disk and memory, can be made brief.

[0138] Moreover, according to the file management equipment and the file management approach concerning this invention, when a hard disk is added in information processing system, such as a computer, the gestalt of the file classification by the existing directory can be followed as it is, and a user can be provided with the user-friendliness that storage capacity increased.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is drawing having shown typically the procedure of the file management performed when extending a storage and saving a new file.

[Drawing 2] It is drawing having shown typically the procedure of the file management performed when saving the file size increased and it became impossible to save by an editing task etc. at the original storage on another storage.

[Drawing 3] It is drawing having shown typically the procedure of the file management performed when creating the directory it became impossible to create to the original storage on another storage.

[Drawing 4] When carrying out file reference ranging over two or more storages, it is drawing having shown the procedure of the file management to perform typically.

[Drawing 5] When file management is performed according to this invention, it is drawing having shown typically the directory structure reflected in a user.

[Drawing 6] It is drawing having shown typically the hardware configuration of the computer system which can realize this invention.

[Description of Notations]

100 -- Computer system

101 -- CPU, 102 -- Memory

103 -- Display controller

104 -- User input interface

105 -- Network interface

107 -- An external instrument interface, 108 -- Bus

111 -- A display, 112 -- A keyboard, 113 -- Mouse

150 -- Storage

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-244901
(P2002-244901A)

(43) 公開日 平成14年8月30日 (2002.8.30)

(51) Int. Cl. ⁷	識別記号	F I	特許出願公開番号 (参考)
G 0 6 F 12/00	5 2 0	G 0 6 F 12/00	5 2 0 P 5 B 0 7 5 5 2 0 J 5 B 0 8 2 6 0 1 S 1 5 0 B
17/30	5 0 1 1 5 0	17/30	

審査請求 未請求 請求項の数11 O L (全 21 頁)

(21) 出願番号 特願2001-41516 (P2001-41516)

(22) 出願日 平成13年2月19日 (2001.2.19)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 中橋 英通

東京都品川区東五反田1丁目14番10号 株式会社ソニー木原研究所内

(74) 代理人 100101801

弁理士 山田 英治 (外2名)

Fターム(参考) 5B075 ND35

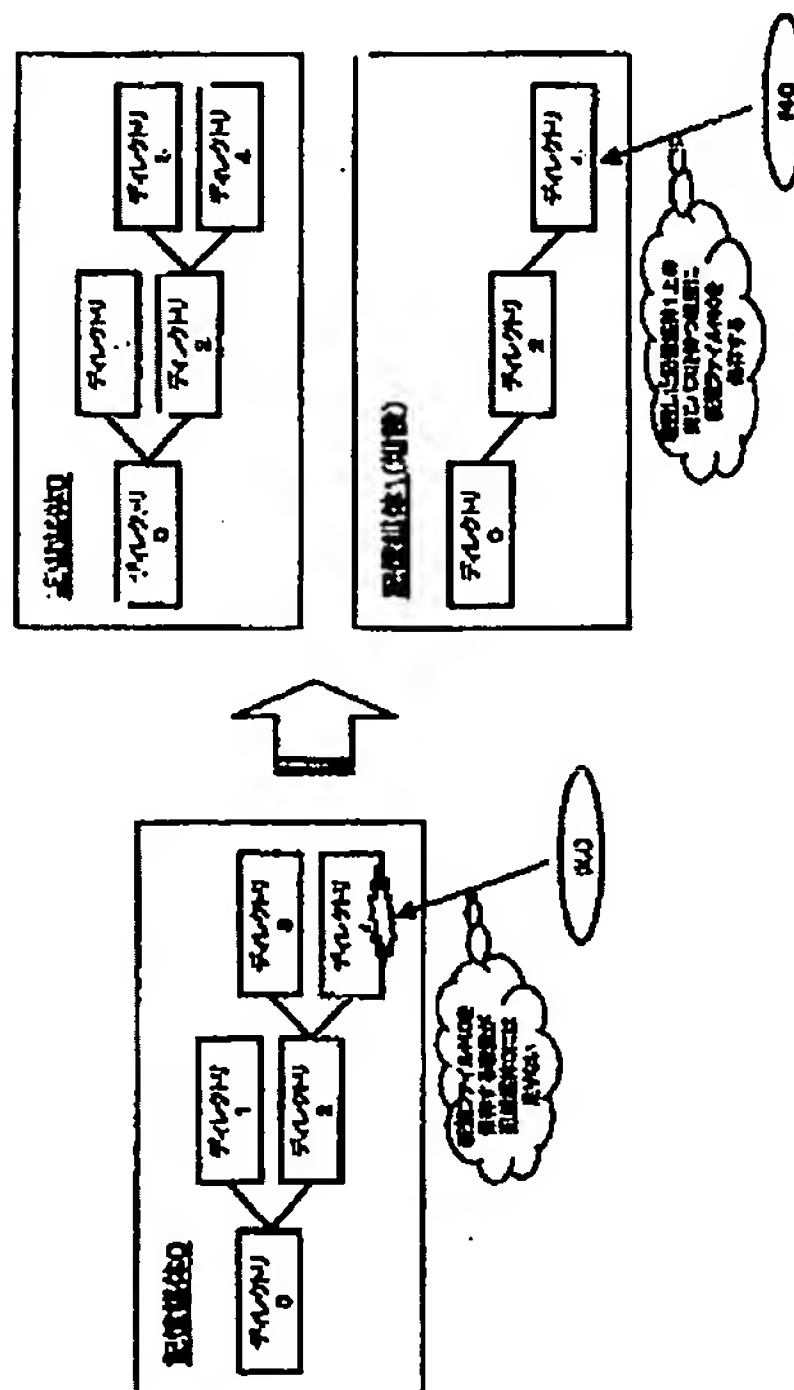
5B082 CA17 EA01

(54) 【発明の名称】 ファイル管理装置及びファイル管理方法、並びに記憶媒体

(57) 【要約】

【課題】 ユーザから見たディレクトリの木構造を変化させずに記憶媒体を追加して記憶容量を増加させることができる、ファイル記憶機能を提供する。

【解決手段】 あるファイルを格納しようとして記憶媒体の容量が足りなかったとき、その媒体上でのファイル格納場所へのパスと同じパスを増設した記憶媒体上に作り、そこに当該ファイルを格納する。また、既存のファイルを編集することでサイズが大きくなり今までの記憶媒体に格納できなくなった場合、容量が空いている別の媒体上に今までの格納先であったディレクトリまでのパスを生成して、そこに当該ファイルを格納する。



【特許請求の範囲】

【請求項1】複数の記憶媒体からなる記憶空間に木構造を導入して階層的にファイルを管理するためのファイル管理装置であって、

所定の木構造が形成された第1の記憶媒体と、

空き容量が確保された第2の記憶媒体と、

前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成手段と、

該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存手段と、を具備することを特徴とするファイル管理装置。

【請求項2】前記パス生成手段は、前記第1の記憶媒体にファイルを保存すべき空き容量がなかったことに応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項1に記載のファイル管理装置。

【請求項3】各記憶媒体上のファイルを編集するファイル編集手段をさらに備え、

前記パス生成手段は、前記第1の記憶媒体上のファイルを編集してファイル・サイズが増大した結果として前記第1の記憶媒体上に保存すべき空き容量がなくなったことに応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項1に記載のファイル管理装置。

【請求項4】各記憶媒体上の木構造を作成する木構造作成手段をさらに備え、

前記パス生成手段は、前記第1の記憶媒体上に木構造を作成するための空き容量がなくなったことに応答して、該作成すべき木構造に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項1に記載のファイル管理装置。

【請求項5】各記憶媒体上で同じパス名と同じファイル名のファイルが存在するか否かを探索するファイル探索手段をさらに備える、ことを特徴とする請求項1に記載のファイル管理装置。

【請求項6】所定の木構造が形成された第1の記憶媒体と空き容量が確保された第2の記憶媒体を含む複数の記憶媒体からなる記憶空間上で階層的にファイルを管理するためのファイル管理方法であって、
前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成ステップと、
該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存ステップと、を具備することを特徴とするファイル管理方法。

【請求項7】前記パス生成ステップでは、前記第1の記憶媒体にファイルを保存すべき空き容量がなかったこと

に応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項6に記載のファイル管理方法。

【請求項8】各記憶媒体上のファイルを編集するファイル編集ステップをさらに備え、

前記パス生成ステップでは、前記第1の記憶媒体上のファイルを編集してファイル・サイズが増大した結果として前記第1の記憶媒体上に保存すべき空き容量がなくなったことに応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項6に記載のファイル管理方法。

【請求項9】各記憶媒体上の木構造を作成する木構造作成ステップをさらに備え、

前記パス生成ステップでは、前記第1の記憶媒体上に木構造を作成するための空き容量がなくなったことに応答して、該作成すべき木構造に到達するための同じパスを前記第2の記憶媒体上に生成する、ことを特徴とする請求項6に記載のファイル管理方法。

【請求項10】各記憶媒体上で同じパス名と同じファイル名のファイルが存在するか否かを探索するファイル探索ステップをさらに備える、ことを特徴とする請求項6に記載のファイル管理方法。

【請求項11】所定の木構造が形成された第1の記憶媒体と空き容量が確保された第2の記憶媒体を含む複数の記憶媒体からなる記憶空間上で階層的にファイルを管理するための処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、
前記コンピュータ・ソフトウェアは、
前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成ステップと、
該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存ステップと、を具備することを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・システム上で取扱い可能にデジタル化された多数のファイルを管理するためのファイル管理装置及びファイル管理方法に係り、特に、ディレクトリなどの木構造を記憶空間に導入して階層的にファイルを管理するためのファイル管理装置及びファイル管理方法に関する。

【0002】更に詳しくは、本発明は、ハード・ディスクなどの複数の記憶媒体を追加可能に構成されたコンピュータ上で、ディレクトリ構造を用いて多数のファイルを管理するファイル管理装置及びファイル管理方法に係り、特に、ユーザから見たディレクトリの木構造を変化

させずに記憶媒体を追加して記憶容量を増加させるファイル管理装置及びファイル管理方法に関する。

【0003】

【従来の技術】昨今の技術革新に伴い、各種の情報処理・情報通信機器の普及が急速に進み、オフィスや一般家庭などに深く浸透してきている。これと相俟って、単なるテキスト・データのみならず、画像や音声などさまざまなデータはデジタル化され、いわゆる「ファイル」という形式でコンピュータ上で取り扱われるようになってきた。

【0004】コンピュータ・ファイルは、一般に、オペレーティング・システム内のファイル・システムの管理下で、メモリやハード・ディスク、フロッピー（登録商標）・ディスク、あるいはその他各種の記録メディア上に保存される。また、ファイル・システムは、「ディレクトリ」と呼ばれる木構造を導入することによって、多数のファイルを階層的に管理できるようになっている。

【0005】一般的なファイル・システムでは、ハード・ディスク・ドライブ、フロッピー・ディスク・ドライブ、CD-ROMドライブなどのように、メディア単位又はドライブ単位でそれぞれ独立したディレクトリの木構造が形成される。したがって、ハード・ディスクなどのある特定のドライブ内で形成されたディレクトリ構造が便利である場合において、他のドライブ又は他のメディアにも同様のディレクトリ構造を適用した場合には、ユーザ自らのシステム操作によって他のドライブ上にディレクトリ構造を改めて作成しなければならない。

【0006】他方、コンピュータ・システムは、コストやファイル空間の制限、その他の設計的な事情により、標準装備される記憶容量には限界がある。このため、ほとんどすべてのシステムにおいては、ハード・ディスクなどの記憶媒体を追加可能な記憶機能を装備している。

【0007】増設したハード・ディスクにおいて好適にファイル管理を行うためには、新たにディレクトリ構造を決定しなければならない。例えば米マイクロソフト社の「Windows」に代表されるオペレーティング・システムでは、増設したハード・ディスク上で新たなディレクトリ構造を作り、そこにファイルを格納していくことになる。

【0008】このような手法では、ユーザがファイルの分類・整理のためにディレクトリ構造を作り、そこにファイルを格納しておいても、さらに新しいハード・ディスクを使用するときには、いままで使用していたハード・ディスク上の既存のディレクトリ以外に別のディレクトリを逐次作って、その新規のファイルを格納することになる。ハード・ディスクの容量が不足して、新たなハード・ディスクを追加する度に、このような煩わしさに苛まれることになる。

【0009】このようなユーザのマニュアル操作に委ねられたファイル管理方法では、同じ分類に属し本来は同

じディレクトリ内に格納されるべきファイルどうしが、ファイル作成時期又は保存時期によっては、異なるハード・ディスク又は異なるディレクトリ内に格納されてことになる。すなわち、ハード・ディスク増設や新規ディレクトリ作成などが間に挿入される結果として、同じ分類に属すべきファイルが複数のディレクトリに分散してしまう。このような場合、後日、一方のファイルを閲覧又は編集している際に、関連のあるファイルを参照しなくなったとしても、同じディレクトリにないがゆえに引き出すことができないこともある。さらに、格納場所を思い出すことができず、ファイル空間の中に埋もれて行方不明のままになってしまいかねない。

【0010】また、開発環境用又はサーバ用として当業界において広く知られているUnix系のオペレーティング・システムにおいては、マウント（Mount：ハード・ディスクなどの周辺機器をコンピュータ上でアクセスできるようにすること）によって1つのディレクトリ構造をなし、デスクトップ上でのファイル操作性を向上させているが、それでも新たなディレクトリが発生することには変わりはない。シンボリック・リンクにより整理することができる。しかしながら、ユーザが明示的なリンクを張る作業が必要なことと、実際（物理的）にディレクトリ構造が変化することを考慮すると、ユーザにとって煩雑であることに変わりはない。

【0011】現在のコンピュータ・システムでは、ハード・ディスクの増設などによりほぼ無限のファイル記憶空間を提供することができるが、これは単にシステムの記憶容量が増えるという効果でしかない。すなわち、デスクトップ画面上で操作するユーザの立場で見ると、追加された記憶媒体はシステムの記憶容量を増やすだけであり、ディレクトリによるファイル分類には寄与しない。ディスクを増設すると、そのディスクに付随した木構造を、今までとは別に独立に作る必要がある。増設したディスク上で作成されたディレクトリは、増設前の木構造とは並存し、ファイルの仕分けが煩雑になる。

【0012】

【発明が解決しようとする課題】本発明の目的は、ディレクトリなどの木構造を記憶空間に導入して階層的にファイルを管理することができる、優れたファイル管理装置及びファイル管理方法を提供することにある。

【0013】本発明の更なる目的は、ハード・ディスクなどの記憶媒体を追加可能に構成されたコンピュータ上で、ディレクトリ構造を用いて多数のファイルを管理することができる、優れたファイル管理装置及びファイル管理方法を提供することにある。

【0014】本発明の更なる目的は、ユーザから見たディレクトリの木構造を変化させずに記憶媒体を追加して記憶容量を増加させることができる、優れたファイル管理装置及びファイル管理方法を提供することにある。

【0015】

【課題を解決するための手段及び作用】本発明は、上記課題を参酌してなされたものであり、その第1の側面は、複数の記憶媒体からなる記憶空間に木構造を導入して階層的にファイルを管理するためのファイル管理装置であって、所定の木構造が形成された第1の記憶媒体と、空き容量が確保された第2の記憶媒体と、前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成手段と、該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存手段と、を具備することを特徴とするファイル管理装置である。

【0016】また、本発明の第2の側面は、所定の木構造が形成された第1の記憶媒体と空き容量が確保された第2の記憶媒体を含む複数の記憶媒体からなる記憶空間上で階層的にファイルを管理するためのファイル管理方法であって、前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成ステップと、該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存ステップと、を具備することを特徴とするファイル管理方法である。

【0017】前記パス生成手段又はステップは、前記第1の記憶媒体にファイルを保存すべき空き容量がなかったことに応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成することができる。

【0018】したがって、本発明の第1又は第2の側面に係るファイル管理装置又は方法によれば、あるファイルを格納しようとして記憶媒体の容量が足りなかったとき、その媒体上でのファイル格納場所へのパスと同じパスを増設した記憶媒体上に作り、そこに当該ファイルを格納することができる。

【0019】すなわち、本発明の第1又は第2の側面によれば、例えば記憶媒体を増設した後であっても、これまでと同じパスで参照可能な状態で新規ファイルが格納される。ユーザは、例えばGUI (Graphical User Interface) 環境が提供されたデスクトップ画面上では、ファイルを該当するディレクトリ内に置くというマウスのドラッグ・アンド・ドロップ操作を行うだけでよいので、ファイル操作性が簡潔である。

【0020】また、本発明の第1又は第2の側面に係るファイル管理装置又は方法は、各記憶媒体上のファイルを編集するファイル編集手段又はステップをさらに備えていてもよい。このような場合、前記パス生成手段又はステップは、前記第1の記憶媒体上のファイルを編集してファイル・サイズが増大した結果として前記第1の記憶媒体上に保存すべき空き容量がなくなったことに応答して、前記木構造上の該ファイルを配置すべき位置に到達するための同じパスを前記第2の記憶媒体上に生成するようにすればよい。

【0021】すなわち、本発明の第1又は第2の側面によれば、編集作業によってサイズが大きくなったファイルがこれまでと同じパスで参照可能な状態で他の記憶媒体上に格納される。ユーザは、例えばGUI環境が提供されたデスクトップ画面上では、ファイルを該当するディレクトリ内に置くというマウスのドラッグ・アンド・ドロップ操作を行うだけでよいので、ファイル操作性が簡潔である。

【0022】また、本発明の第1又は第2の側面に係るファイル管理装置又は方法は、各記憶媒体上の木構造を作成する木構造作成手段又はステップをさらに備えていてもよい。このような場合、前記パス生成手段又はステップは、前記第1の記憶媒体上に木構造を作成するための空き容量がなくなったことに応答して、該作成すべき木構造に到達するための同じパスを前記第2の記憶媒体上に生成するようにすればよい。

【0023】すなわち、本発明の第1又は第2の側面によれば、これまでと同じパスで参照可能な状態で新規のディレクトリが別の記憶媒体上に生成される。ユーザは、例えばGUI環境が提供されたデスクトップ画面上では、各記憶媒体の区別を意識することなく、通常のマウス操作に従って新規のディレクトリにアクセスすることができるので、ファイル操作性が簡潔である。

【0024】また、本発明の第1又は第2の側面に係るファイル管理装置又は方法は、各記憶媒体上で同じパス名と同じファイル名のファイルが存在するか否かを探索するファイル探索手段又はステップをさらに備えていてもよい。

【0025】このファイル探索手段又はステップによれば、記憶媒体を増設したとしても、これまでと同じパスで参照可能な状態でいかなるファイルも読み出すことができる。ユーザは、例えばGUI環境が提供されたデスクトップ画面上では、所望のファイルがいずれの記憶媒体に配置されているかを意識することなく、通常のマウス操作に従ってファイルにアクセスすることができるので、ファイル操作性が簡潔である。

【0026】しかして、本発明に係るファイル管理手法に従えば、ユーザからは、システムが提供する単一の記憶領域上に統一的なディレクトリ構造が形成されているように見える。したがって、記憶媒体が追加されても、ユーザからは記憶領域の容量が単純に増えたとしか見えず、いままでのディレクトリ構造だけを意識してファイル操作を行うことができる。

【0027】記憶媒体の増設の繰り返しにより各ディレクトリは物理的には異なる記憶媒体上に分散して配置されている。本発明によれば、ディレクトリに対しても、これまでと同じパスで参照可能な状態でファイルが格納される。ユーザは、例えばGUI環境が提供されたデスクトップ画面上では、マウスのドラッグ・アンド・ドロップ操作を行うだけで、ファイルを所望のディレクトリ

に保存したり、読み出したり、移動したり、削除したりといった通常のファイル操作を実行することができるので、ファイル操作性が簡潔である。

【0028】また、本発明の第3の側面は、所定の木構造が形成された第1の記憶媒体と空き容量が確保された第2の記憶媒体を含む複数の記憶媒体からなる記憶空間上で階層的にファイルを管理するための処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ・ソフトウェアは、前記第1の記憶媒体における木構造上の所定位置に到達するための同じパスを前記第2の記憶媒体上に生成するパス生成ステップと、該生成されたパスを用いてファイルを前記第2の記憶媒体に保存するファイル保存ステップと、を具備することを特徴とする記憶媒体である。

【0029】本発明の第3の側面に係る記憶媒体は、例えば、様々なプログラムコードを実行可能な汎用コンピュータシステムに対して、コンピュータ・ソフトウェアをコンピュータ可読な形式で提供する媒体である。このような媒体は、例えば、CD (Compact Disc) やFD (Floppy Disk)、MO (Magnet-Optical disc) などの着脱自在で可搬性の記憶媒体である。あるいは、ネットワーク（ネットワークは無線、有線の区別を問わない）などの伝送媒体などを經由してコンピュータ・ソフトウェアを特定のコンピュータ・システムに提供することも技術的に可能である。

【0030】このような記憶媒体は、コンピュータ・システム上で所定のコンピュータ・ソフトウェアの機能を実現するための、コンピュータ・ソフトウェアと記憶媒体との構造上又は機能上の協働的關係を定義したものである。換言すれば、本発明の第3の側面に係る記憶媒体を介して所定のコンピュータ・ソフトウェアをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の第1及び第2の各側面に係るファイル管理装置及び方法と同様の作用効果を得ることができる。

【0031】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基くより詳細な説明によって明らかになるであろう。

【0032】

【発明の実施の形態】本発明に係るファイル管理装置又はファイル管理方法は、ハード・ディスクなどの記憶媒体を追加可能に構成されたコンピュータ上で、ユーザから見たディレクトリの木構造を変化させずに記憶媒体を追加して記憶容量を増加させるものであり、例えばデスクトップ上におけるユーザによるファイル操作やファイル分類を簡潔にするものである。ここでは、まず、本発明に係るファイル管理の処理手順について概略的に説明する。

【0033】(1) 新規ファイル保存時

ハード・ディスクなどの記憶媒体上で分類してファイル管理するために、一般に、ディレクトリ構造を作成して、各ファイルを該当するディレクトリ内に保存する。

【0034】あるファイルを格納しようとして記憶媒体の容量が足りなかったとき、新たな記憶媒体を増設する。このとき、元の記憶媒体上でのファイルの格納場所へのパスと同じパスを追加した記憶媒体上にも作成して、そこに当該ファイルを作成する。その手順は記憶媒体を何個追加しても同様である。

【0035】図1には、記憶媒体を増設して新しいファイルを保存するときに実行するファイル管理の手順を模式的に示している。

【0036】同図に示す例では、まず、システムには記憶媒体0が装備されているとともに、この記憶媒体0内には、ルート・ディレクトリ0以下にサブ・ディレクトリ1及び2が形成され、さらにディレクトリ2以下にはそのサブ・ディレクトリ3及び4が形成されている。

【0037】ここで、ディレクトリ4内の新規のファイルf40を保存しようとしたときに、記憶媒体0の空き容量不足で保存できなかったとする。

【0038】このような場合、新規ファイルf40を保存するために、新しい記憶媒体1をシステムを増設する。

【0039】本発明によれば、記憶媒体1上には、ルート・ディレクトリ0以下にサブ・ディレクトリ2を配置するとともに、ディレクトリ2以下にサブ・ディレクトリ4を配置して、元の記憶媒体0上のディレクトリ4と同じパスを新しい記憶媒体1上に用意する。そして、本来は記憶媒体0上のディレクトリ4に保存すべきであったファイルf40を新しい記憶媒体1側のディレクトリ4内に保存する。

【0040】すなわち、本発明によれば、記憶媒体1を増設した後、システムの内部処理によって、これまでと同じパスで参照可能な状態で新規ファイルf40が格納される。ユーザは、例えばGUI (Graphical User Interface) 環境が提供されたデスクトップ画面上では、ファイルf40をディレクトリ4内に置くというマウスのドラッグ・アンド・ドロップ操作を行うだけでよいので、ファイル操作性が簡潔である。

【0041】(2) ファイル編集時

既に存在するファイルを編集することで、そのファイル・サイズが以前よりも大きくなってしまい、元の記憶媒体上には格納する空き容量が残っていないことがある。

【0042】このような場合、容量が空いている別の記憶媒体上に、当該ファイルを格納するディレクトリのパスが既に用意されていれば、そこに保存すればよい。また、パスが未だ用意されていなければ、そのディレクトリまでのパスを生成して、そこに当該ファイルを保存する。もし、そのディレクトリ・パスを生成する容量が別

の記憶媒体上にももはや空いていなければ、さらに別の記憶媒体上で同様の手順に従い処理する。

【0043】図2には、編集作業などによってサイズが増大して、元の記憶媒体に保存できなくなったファイルを別の記憶媒体上に保存するときに実行するファイル管理の手順を模式的に示している。

【0044】同図に示す例では、システムには元から装備された記憶媒体0と、後から追加増設された記憶媒体1が存在する。また、記憶媒体0内には、ルート・ディレクトリ0以下にサブ・ディレクトリ1及び2が形成され、さらにディレクトリ2以下にはそのサブ・ディレクトリ3及び4が形成されている。他方の記憶媒体1上には、ディレクトリ4内のファイルを参照するための記憶媒体0と同じディレクトリ・パス、すなわちディレクトリ0以下にディレクトリ2を配置するとともにディレクトリ2以下にディレクトリ4を配置するというディレクトリ・パスが形成されている。

【0045】ここで、記憶媒体0上でディレクトリ3内のファイルf30を編集したら、記憶媒体0の空き容量不足で保存できなくなったとする。

【0046】このような場合、まず、編集後のファイルf30にアクセスするためのディレクトリ3までのパスが、増設した記憶媒体1上にすで存在するか否かを判別する。もし、このようなディレクトリ・パスが存在していれば、それを利用して記憶媒体1上にファイルf30を保存する。

【0047】図2に示す例では、記憶媒体1上には、ファイルf30にアクセスするためのディレクトリ3までのパスはまだ用意されていない。そこで、記憶媒体1上にも、ルート・ディレクトリ0のサブ・ディレクトリ2以下にディレクトリ3を配置することによって、記憶媒体0と同じディレクトリ・パスを生成して、そのディレクトリ3内にファイルf30を保存する。

【0048】最後に、記憶媒体0のディレクトリ3内にあった元のファイルf30を消去して、記憶媒体1上に新規に保存したファイルf30との競合を回避する。元のファイルの消去は、記憶媒体0の記憶空間の解放にもつながる。

【0049】すなわち、本発明によれば、記憶媒体1を増設した後、システムの内部処理によって、サイズが大きくなったファイルf30がこれまでと同じパスで参照可能な状態で格納される。ユーザは、例えばGUI (Graphical User Interface) 環境が提供されたデスクトップ画面上では、ファイルf30をディレクトリ3内に置くというマウスのドラッグ・アンド・ドロップ操作を行うだけでよいので、ファイル操作性が簡潔である。

【0050】なお、図示しないが、記憶媒体1上にファイルf30にアクセスするためのディレクトリ3までのパスが既に用意されているが、記憶媒体1上にも十分な空き容量がない場合には、十分な空き容量を持つ別の記

憶媒体上で、上記と同じディレクトリ・パスの生成及びファイル保存の作業を実行する。

【0051】(3) 新規ディレクトリ作成時
新しいディレクトリを作成したいときには、そのディレクトリを生成するに十分な記憶容量が空いている記憶媒体上に、そのディレクトリまでのパスを生成する。

【0052】この結果、いずれの記憶媒体上にディレクトリが配置されようとも、同じディレクトリ・パスを持つので、ユーザはディレクトリの物理的な場所を意識することなくディレクトリ内のファイルにアクセスすることができる。

【0053】図3には、元の記憶媒体に作成できなくなったディレクトリを別の記憶媒体上に作成するときに実行するファイル管理の手順を模式的に示している。

【0054】同図に示す例では、システムには元から装備された記憶媒体0と、後から追加増設された記憶媒体1が存在する。また、記憶媒体0内には、ルート・ディレクトリ0以下にサブ・ディレクトリ1及び2が形成され、さらにディレクトリ2以下にはそのサブ・ディレクトリ3及び4が形成されている。他方の記憶媒体1上には、ディレクトリ4内のファイルを参照するための記憶媒体0と同じディレクトリ・パス、すなわちディレクトリ0以下にディレクトリ2を配置するとともにディレクトリ2以下にディレクトリ4を配置するというディレクトリ・パスが形成されている。

【0055】ここで、記憶媒体0上で、ディレクトリ2以下に新規のディレクトリ5を生成しようとするが、記憶媒体0の記憶容量不足でディレクトリを生成することができなかったとする。

【0056】このような場合、まず、増設した記憶媒体1上に新規のディレクトリを生成するために十分な空き容量があるか否かを判別する。そして、もし十分な空き容量があれば、記憶媒体1上で、同じパスを持つ場所すなわちディレクトリ2以下にディレクトリ5を生成する。

【0057】すなわち、本発明によれば、記憶媒体1を増設した後、システムの内部処理によって、これまでと同じパスで参照可能な状態で新規のディレクトリ5が生成される。ユーザは、例えばGUI (Graphical User Interface) 環境が提供されたデスクトップ画面上では、記憶媒体0又は記憶媒体1のいずれに配置されているかを意識することなく、通常のマウス操作に従って新規のディレクトリ5にアクセスすることができるので、ファイル操作性が簡潔である。

【0058】なお、図示しないが、記憶媒体1上にも十分な空き容量がない場合には、十分な空き容量を持つ別の記憶媒体上で、上記と同じパスを持つ場所に新規ディレクトリを生成する。

【0059】(4) ファイル読出し時
ファイルを読み出すときは、パス名とファイル名を基

に、各記憶媒体を順番に探索する。

【0060】新たなファイルがあるバスのディレクトリに生成するときは、同じバス名とファイル名のファイルがあるかどうかを、記憶媒体を順番に探索して、ファイルの上書きなどの問題が出現しないようにする。

【0061】記憶媒体を探索する順番は、例えば、標準装備された記憶媒体を先頭にして、以下、増設された順番であってもよい。

【0062】図4には、複数の記憶媒体にまたがってファイル参照するときに実行するファイル管理の手順を模式的に示している。

【0063】同図に示す例では、システムには元から装備された記憶媒体0と、後から追加増設された記憶媒体1及び記憶媒体2が存在する。記憶媒体0内には、ルート・ディレクトリ0以下にサブ・ディレクトリ1及び2が形成され、さらにディレクトリ2以下にはそのサブ・ディレクトリ3及び4が形成されている。また、記憶媒体1上には、ディレクトリ4内のファイルを参照するための記憶媒体0と同じディレクトリ・パス、すなわちディレクトリ0以下にディレクトリ2を配置するとともにディレクトリ2以下にディレクトリ4を配置するというディレクトリ・パスが形成されている。また、記憶媒体2上には、ディレクトリ5内のファイルを参照するための記憶媒体0と同じディレクトリ・パス、すなわちディレクトリ0以下にディレクトリ2を配置するとともにディレクトリ2以下にディレクトリ5を配置するというディレクトリ・パスが形成されている。

【0064】ここで、システムのファイル空間上で、ルート・ディレクトリ0以下のサブ・ディレクトリ2以下のサブ・ディレクトリ5内に保存されているファイルf51、すなわち“Dir0/Dir2/Dir5/f51”を読み出す旨のコマンドが発行されたとする。

【0065】このような場合、まず、記憶媒体0上のルート・ディレクトリ0以下のサブ・ディレクトリ2以下にディレクトリ5が存在するか否か、木探索を行う。同図に示す例では、記憶媒体0上にはディレクトリ5は存在しない。そこで、増設した記憶媒体1上でも、ルート・ディレクトリ0以下のサブ・ディレクトリ2以下にディレクトリ5が存在するか否か、木探索を行う。同図に示す例では、記憶媒体1上にもディレクトリ5は存在しない。そこで、増設した記憶媒体2上でも、ルート・ディレクトリ0以下のサブ・ディレクトリ2以下にディレクトリ5が存在するか否か、さらに木探索を行う。そして、記憶媒体2上でディレクトリ5を成功裏に発見することができるので、そのディレクトリ内で読み出し対象となっているファイルf51に読み出しアクセスする。

【0066】すなわち、本発明によれば、記憶媒体1を増設した後、システムの内部処理によって、これまでと同じバスで参照可能な状態でいかなるファイルも読み出すことができる。ユーザは、例えばGUI (Graphical

User Interface) 環境が提供されたデスクトップ画面上では、所望のファイルが記憶媒体0、あるいは増設した記憶媒体1、記憶媒体2のいずれに配置されているかを意識することなく、通常のマウス操作に従ってファイルにアクセスすることができるので、ファイル操作性が簡潔である。

【0067】図5には、本発明に従ってファイル管理を行った場合に、ユーザに映るディレクトリ構造を模式的に示している。

【0068】システムには実際には複数の独立した記憶媒体が共存し、且つ、各ディレクトリは物理的には2以上の記憶媒体にまたがって分散して配置されている。しかしながら、上述した本発明に係るファイル管理手法に従えば、ユーザからは、システムが提供する単一の記憶領域上に統一的なディレクトリ構造が形成されているように見える。

【0069】図5に示す例で言えば、ユーザからは、ルート・ディレクトリ0以下にサブ・ディレクトリ1及びサブ・ディレクトリ2が配置され、さらに、サブ・ディレクトリ2以下にはサブ・ディレクトリ3、サブ・ディレクトリ4、並びにサブ・ディレクトリ5が配置されている。

【0070】記憶媒体の増設の繰り返しにより各ディレクトリは物理的には異なる記憶媒体上に分散して配置されているが、本発明によれば、システムの内部処理によっていかなるディレクトリに対しても、これまでと同じバスで参照可能な状態でファイルが格納される。ユーザは、例えばGUI (Graphical User Interface) 環境が提供されたデスクトップ画面上では、マウスのドラッグ・アンド・ドロップ操作を行うだけで、ファイルを所望のディレクトリに保存したり、読み出したり、移動したり、削除したりといった通常のファイル操作を実行することができるので、ファイル操作性が簡潔である。

【0071】以上を総括すると、本発明に係るファイル管理装置並びにファイル管理方法によれば、システム上に記憶媒体が何個存在するかに拘わらず、ユーザからは1つの記憶領域があるように見える。したがって、記憶媒体が追加されても、ユーザからは記憶領域の容量が単純に増えたとしか見え、いままでのディレクトリ構造だけを意識してファイル操作を行うことができる。

【0072】次いで、本発明の実施形態について具体的に説明する。

【0073】本発明を実現するために、専用のハードウェア構造を備えたファイル管理装置を設計することもできるが、多種類のアプリケーションを実行可能な汎用性のコンピュータ・システム上で所定のプログラム・コードを実行するという形態で、本発明を実装することも可能である。図6には、本発明に係るファイル管理を実現可能なコンピュータ・システム100のハードウェア構成を模式的に示している。以下、同図に従って各ブロッ

クについて説明する。

【0074】システム100のメイン・コントローラであるCPU (Central Processing Unit) 101は、オペレーティング システム (OS) の制御下で、各種のアプリケーションを実行する。図示の通り、CPU 101は、バス108によって他の機器類(後述)と相互接続されている。

【0075】メモリ102は、CPU 101において実行されるプログラム コードを格納したり、実行中の作業データを一時保管するために使用される記憶装置である。また、メモリ102は、ファイル管理又はファイル保存用の記憶媒体としても機能することができる。

【0076】ディスプレイ・コントローラ103は、CPU 101が発行する描画命令を実際に処理するための専用コントローラである。ディスプレイ コントローラ103において処理された描画データは、例えばフレーム・バッファ(図示しない)に一旦書き込まれた後、ディスプレイ111によって画面出力される。

【0077】ユーザ入力インターフェース104は、キーボード (KBD) 112やマウス113などのユーザ入力機器をコンピュータ・システム100に接続するための装置である。

【0078】例えば、ディスプレイ111の画面上で提供される「デスクトップ」とこれらユーザ入力装置112, 113との組み合わせにより、対話型のグラフィック・ベースのユーザ入力環境、GUI (Graphical User Interface) すなわちが提供される。このような場合、システム100のユーザは、デスクトップ上で、それぞれのファイルやアプリケーションに割り付けられたアイコン(絵文字)に対して、マウス113を介してクリックやドラッグ・アンド・ドロップといった操作を印加することによって、対応するファイルやアプリケーションに対するユーザ・コマンドを直感的に入力することができる。勿論、ディレクトリへのファイルの保存や、ファイル読出し、ディレクトリ間のファイル移動なども、マウス操作を基調として実行される。

【0079】ネットワーク インターフェース105は、Ethernetなどの所定の通信プロトコルに従って、システム100をLAN (Local Area Network) などの局所的ネットワーク、さらにはインターネットのような広域ネットワークに接続することができる。

【0080】ネットワーク上では、コンテンツ受信装置などの複数のホスト端末がトランスペアレントな状態で接続され、分散コンピューティング環境が構築されている。例えば、ネットワーク上の所定のサーバから、ファイルやさまざまなコンテンツをダウンロードすることができる。勿論、本発明に係るファイル管理を実現するためのコンピュータ・ソフトウェアもネットワーク経由でダウンロードすることができる。

【0081】外部機器インターフェース107は、2以

上の記憶媒体150A, 150B, ...を本コンピュータ・システム100に接続するための装置である。外部機器インターフェース107は、例えばIDE (Integrated Drive Electronics) やSCSI (Small Computer System Interface) などのインターフェース仕様に準拠して設計・製作されている。

【0082】それぞれの記憶媒体150A, 150B...上には、ファイル形式で各種のデータやコンテンツを格納することができる。記憶媒体150A, 150B...のうち少なくとも一部はコンピュータ・システム100に標準装備され、他は増設される。増設可能な記憶媒体の最大個数は、外部機器インターフェース107に採用されたインターフェース仕様に応じて決定される。記憶媒体150A, 150B...は、例えばハード ディスク ドライブ、MO (Magnet-Optical disc) ドライブ、DVD (Digital Versatile) -RAMドライブなど、読み書き可能な記憶媒体で構成される。勿論、CD-ROMドライブやDVD-ROMドライブのように、読出し専用の記憶媒体を一部に含んでいてもよい。

【0083】ハード・ディスク・ドライブは、記憶担体としての磁気ディスクを固定的に搭載した外部記憶装置であり(周知)、記憶容量やデータ転送速度などの点で他の外部記憶装置よりも優れている。ソフトウェア・プログラムを実行可能な状態でハード・ディスク上に置くことをプログラムのシステムへの「インストール」と呼ぶ。通常、ハード・ディスクには、CPU 101が実行すべきプログラム コードが不揮発的に格納されている。例えば、本発明の実施形態に係るファイル管理を行うソフトウェアをハード・ディスク上にインストールすることができる。勿論、このファイル管理下に置かれる複数のファイルをハード・ディスク上に保存することもできる。

【0084】記憶媒体が、CD (Compact Disc) やMO (Magnet-Optical disc)、DVD (Digital Versatile Disc) などの交換可能な可搬型メディアで構成される場合、ソフトウェア プログラムやデータ ファイルなどをコンピュータ可読形式のデータとしてバックアップすることや、これらをシステム間で移動(すなわち販売流通 配布を含む)する目的で使用される。例えば、本発明の実施形態に係るファイル管理を行うコンピュータ・ソフトウェアを、これら可搬型メディアを利用して複数の機器間で物理的に流通・配布することができる。勿論、このファイル管理下に置かれる複数のファイルを、可搬型メディアを利用して機器間で物理的に流通・配布することもできる。

【0085】なお、図6に示すようなコンピュータ・システム100の一例は、米IBM社のパーソナル コンピュータ"PC/AT (Personal Computer/Advanced Technology)"の互換機又は後継機である。勿論、他のアーキテクチャを備えたコンピュータを、本実施形態に係

るコンピュータ・システム100として適用することも可能である。

【0086】次いで、上述したコンピュータ・システム100上において実現されるファイル管理方法について説明する。

【0087】1. 実施形態1

コンピュータ・システム100においては、一般に、各周辺機器毎にデバイス・ドライバが用意されており、オペレーティング・システムなどの上位ソフトウェアは、デバイス・ドライバを介して対応する機器操作を制御することによって、システム全体を統括する。勿論、上述した各記憶媒体150A、150B…毎にデバイス・ドライバが配置されている。

【0088】本実施形態では、コンピュータ・システム100に装備された1つの記憶媒体150Aは特別な意味を持ち、これには、システム100内におけるファイルの格納情報、その木構造の情報を管理するLUT (Look Up Table) を書き込むという目的が与えられている。このLUTに書かれている木構造は、システムにただ1つ存在する木構造（本明細書中では、以下「ツリー0」とする）である。各記憶媒体150A、150B…はそれぞれ木構造を持つが、それぞれの木構造はシステム全体の木構造すなわちツリー0の部分集合であると把握されたい。

【0089】各記憶媒体150A、150B…が持つ木構造と、システム全体の木構造「ツリー0」との関係は以下の通りである。すなわち、あるファイルF0を/d0/d1/d2/d3というディレクトリに格納するとき、ユーザには見えない（あるいは、デスクトップ上では可視化されていない）がK番目の記憶媒体にシステムが格納しようとしているとする。このとき、K番目の記憶媒体には/d0/d1/d2/d3というディレクトリが存在しなかったならば、まず、このK番目の記憶媒体に/d0/d1/d2/d3というディレクトリを作成する。そして、そこにファイルF0を格納する。このようなシステム100の内部的なコントロールがなされ

N=システムの格納用記憶媒体の個数

K=1

LOOP1:

IF (K番目の記憶媒体の木構造に/d0/d1/d2が存在する)

{

IF (K番目の記憶媒体にF0を格納する空き容量がある) {
K番目の記憶媒体の/d0/d1/d2にF0を格納する
LUTに/d0/d1/d2/F0についての情報を書き込む
GOTO END

}

ELSE {
IF (K=N) { GOTO ERROR }
ELSE { K=K+1 GOTO LOOP1 }

}

ていった結果、各記憶媒体には、システム全体の木構造であるツリー0の部分照合としての独自の木構造が構築されていく。

【0090】本発明の第1の実施形態では、LUTには、システム100上のファイル格納情報が書かれる。書かれている情報としては、木構造を表す情報、それぞれのディレクトリに格納されているファイル名、そのファイルが実際に格納されている記憶媒体の番号（又は識別情報）、システム100内に装備された記憶媒体の個数などである。GUI環境下において、デスクトップ上でディレクトリ構造やファイル一覧を表示するときにLUTが使用される。また、LUTは、ファイル読出し時の検索の情報源として使用される。また、ファイルの格納、移動、コピー、削除などの操作を行ったときに、LUTの情報は逐次書き換えられる。

【0091】以下では、本発明の第1の実施形態におけるファイル管理の具体的な処理手順について説明する。

【0092】(1) 新規ファイルの格納

ユーザの視点、若しくはデスクトップ上で、新規ファイルF0をあるディレクトリ、例えば/d0/d1/d2に格納しようとする。システム100の全格納容量に十分な空きがあれば、ファイルF0は/d0/d1/d2内にそのまま格納される。他方、システム100の全格納容量に十分な空きがなければ、エラー状態になり、ファイルF0が格納不能である旨のメッセージがユーザに通知される（例えば、デスクトップ上にダイアログが出現する）。

【0093】新規ファイルを格納するためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDOFLOOP*”で挟まれる部分は(“*”は数字)、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0094】

【数1】

```

    }
    ELSE {
        IF ( K番目の媒体に/d0/d1/d2というディレクトリを作りF0を格納
        する空き容量がある ) {
            K番目の媒体に/d0/d1/d2を作る
            K番目の媒体の/d0/d1/d2にF0を格納する
            LUTに/d0/d1/d2/F0についての情報を書き込む
            GOTO END
        }
        ELSE {
            IF ( K=N ) { GOTO ERROR }
            ELSE { K=K+1 GOTO LOOP1 }
        }
    }
ENDofLOOP1:
ERROR:
    もう格納できない旨のメッセージを発行
END:

```

【0095】(2) 既存のファイルを編集・保存
 ユーザの視点、若しくはデスクトップ上で、例えば/d
 3/d4というディレクトリに存在するファイルF1を
 編集して、同じパス名とディレクトリ名で保存しようと
 する。このとき、システム100の全格納容量に十分な
 空きがあれば、ファイルF1は/d3/d4/F1とし
 て上書きされる。他方、システム100の全格納容量に
 十分な空きがなければ、エラー状態になり、ファイルF
 1が格納不能である旨のメッセージがユーザに通知され
 る(例えば、デスクトップ上にダイアログが出現す

る)。

【0096】既存ファイルを編集後に保存するためのシ
 ステム内部の動きを、擬似コード形式で以下に示す。但
 し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は(“*”は
 数字)、“goto”文により抜け出さない限り、繰り返し実
 行されるループ処理である。また、“IF”で始まる文は、
 条件判断文であり、それが満たされれば“{”で囲まれた
 記述内容に従った動作を実行する。

【0097】

【数2】

```

N=システムの格納用記憶媒体の個数
M=編集前、ファイルF1が格納されていた記憶媒体の番号
F1の編集後、
IF ( M番目の記憶媒体に/d3/d4/F1/を上書きするための容量が充分空いている )
{
    M番目の記憶媒体の/d3/d4/にF1を格納する
    LUTに/d3/d4/F1/についての情報を書き込む
    GOTO END
}
ELSE {
    K=1
    LOOP2:
        IF ( K=M&&K=N ) { GOTO ERROR }
        ELSE IF ( K=M&&K!=N ) { K=K+1; GOTO LOOP2 }
        ELSE {
            IF ( K番目の記憶媒体の木構造に/d3/d4が存在する )
            {
                IF ( K番目の媒体にF1を格納する空き容量がある )
                {
                    K番目の媒体の/d3/d4にF1を格納する
                    LUTに/d3/d4/F1についての情報を書き込む
                    GOTO END
                }
            }
        }
    }
}

```

```

    }
    ELSE {
        IF (K==N) { GOTO ERROR }
        ELSE { K=K+1 GOTO LOOP2 }
    }
}
ELSE {
    IF (K番目の媒体に/d3/d4というディレクトリを作
りF1を格納する空き容量がある) {
        K番目の媒体の/d3/d4を作る
        K番目の媒体の/d3/d4/にF1を格納する
        LUTに/d3/d4/F1についての情報を書き込む
        GOTO END
    }
    ELSE {
        IF (K==N) { GOTO ERROR }
        ELSE { K=K+1 GOTO LOOP1 }
    }
}
}
}
ENDOFLOOP2:
}

```

ERROR:

格納できない旨のメッセージを発行

END:

【0098】(3) ファイルの読出し
ユーザの視点、若しくはデスクトップ上で、例えば/d5/d6/d7/d8というディレクトリに存在するファイルF2を読み出す操作を印加する。システム100の記憶空間内で、/d5/d6/d7/d8/F2が存在すれば、それを読み出す。なければエラー状態になり、そのパスにF2が存在しないというメッセージがユーザに通知される(例えば、デスクトップ上にダイアログが出現する)。

LUTを参照する

```

IF (LUTのエントリに/d5/d6/d7/d8/F2に関するエントリがある) {
    それが実際に格納されている記憶媒体の番号をLUT参照により知る
    (M番目の媒体だったとする)
    M番目の媒体から/d5/d6/d7/d8/F2を読み出す
    GOTO END
}
ELSE {
    GOTO ERROR
}

```

ERROR: ファイルが存在しない旨のメッセージを発行する

END:

【0101】(4) ファイルのコピー
ユーザの視点、若しくはデスクトップ上で、例えば/d5/d6/d7/d8というディレクトリに存在するファイルF2を/d9というディレクトリにコピーする操

【0099】システム100の記憶空間からファイルを読み出すためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDOFLOOP*”で挟まれる部分は(“*”は数字)、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0100】

【数3】

作を印加する。システム100の記憶容量に、新たなファイル/d9/F2を格納するための空きがあれば、/d9/F2という複製を格納する。空き容量がなければ、エラー状態になり、空き容量がない旨のメッセージ

がユーザに通知される（例えば、デスクトップ上にダイアログが出現する）。

【0102】システム100の内部では、まず／d5／d6／d7／d8／F2を読み出す。そのときの手順は、上記の『（3）ファイルの読出し』で説明した通りである。

【0103】このファイル読出しに成功したら、次いで、／d9／F2という新規ファイルの格納を行う。その手順は、上記の『（1）新規ファイルの格納』で説明した通りである。

【0104】（5）ファイルの削除
ユーザの視点、若しくはデスクトップ上で、例えば／d10／d11／F3と言うファイルを削除しようとする。そ

LUTを参照する

IF (／d10／d11／F3のエントリが存在した) {

LUTにおける／d10／d11／F3に関するエントリを削除する

／d10／d11／F3が格納されている番号の記憶媒体のドライバに対して、／d1

0／d11／F3の削除命令を発行する

GOTO END

}

ELSE {

GOTO ERROR

}

ERROR: 削除すべきファイルがないというメッセージを発行

END:

【0107】（6）ファイルの移動
ユーザの視点、若しくはデスクトップ上で、例えば／d5／d6／d7／d8／F2というファイルを他のディレクトリ、例えば／d12／d13／に移動する操作を印加する。移動のための作業空間として必要な容量がシステム100に充分空いていなかったらエラーになる。

【0108】システム100の内部では、まず、／d5／d6／d7／d8／F2を、／d12／d13／F2としてコピーする。そのときの手順は、上記の『（4）ファイルのコピー』で説明した通りである。そこで、エラーが発生したら一連の動作を中止する。

【0109】エラーでなければ、／d5／d6／d7／d8／F2を削除する。そのときの手順は、上記の『（5）ファイルの削除』で説明した通りである。

【0110】2. 実施形態2

本実施形態では、システム100に複数存在する記憶媒体を統括的にコントロールするソフトウェアが1つ存在することを想定する。1つの記憶媒体が特別な意味を持ち、これにはシステム100内のファイルの格納情報、木構造の情報を管理するLUT (Look Up Table) を書き込むという目的が与えられている。

【0111】本発明の第2の実施形態では、LUTには、システム100上のファイル格納情報が書かれている。書かれている情報としては、木構造を表す情報、それぞれのディレクトリに格納されているファイル名、そ

のファイルがシステム100の記憶空間に存在すれば削除する。なければ、そのようなファイルは存在しない旨のメッセージをユーザに返す（例えば、デスクトップ上にダイアログが出現する）。

【0105】システム100の記憶空間からファイルを削除するためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は（“*”は数字）、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0106】

【数4】

のファイルが実際に格納されている記憶媒体の番号と、その記憶媒体内でのファイルの物理的なアドレス、それぞれの記憶媒体においてどの領域が空いているかなどの空き容量情報、システム100内に装備された記憶媒体の個数などである。

【0112】以下では、本発明の第2の実施形態におけるファイル管理の具体的な処理手順について説明する。

【0113】（1）新規ファイルの格納

ユーザの視点、若しくはデスクトップ上で、新規ファイルF0をあるディレクトリ、例えば／d0／d1／d2に格納しようとする。システム100の全格納容量に十分な空きがあれば、ファイルF0は／d0／d1／d2内にそのまま格納される。他方、システム100の全格納容量に十分な空きがなければ、エラー状態になり、ファイルF0が格納不能である旨のメッセージがユーザに通知される（例えば、デスクトップ上にダイアログが出現する）。

【0114】新規ファイルを格納するためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は（“*”は数字）、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0115】

【数5】

N=システムの格納用記憶媒体の個数

K=1

LOOP1:

```

IF (K番目の媒体にF0を格納する空き容量がある) {
    K番目の媒体にF0を格納する
    LUTに/d0/d1/d2/F0についてのエントリを登録する
    GOTO END
}

```

```

ELSE {
    IF (K==N) { GOTO ERROR }
    ELSE { K=K+1 GOTO LOOP1 }
}

```

ENDofLOOP1:

ERROR: もう格納できない旨のメッセージを発行

END:

【0116】なお、上記の擬似コード中で、LUTに登録する情報は、実際に記憶されている媒体の番号、その倍対中の格納されたアドレス、占有容量などでは。

【0117】(2) 既存のファイルを編集・保存
 ユーザの視点、若しくはデスクトップ上で、例えば/d3/d4というディレクトリに存在するファイルF1を編集して、同じパス名とディレクトリ名で保存しようとする。このとき、システム100の全格納容量に十分な空きがあれば、ファイルF1は/d3/d4/F1として上書きされる。他方、システム100の全格納容量に十分な空きがなければ、エラー状態になり、ファイルF1が格納不能である旨のメッセージがユーザに通知され

る(例えば、デスクトップ上にダイアログが出現する)。

【0118】既存ファイルを編集後に保存するためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は(“*”は数字)、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{”で囲まれた記述内容に従った動作を実行する。

【0119】

【数6】

N=システムの格納用記憶媒体の個数

M=編集前、ファイルF1が格納されていた記憶媒体の番号

F1の編集後、

IF (M番目の記憶媒体に/d3/d4/F1/を上書きするための容量が充分空いている)

```

{
    M番目の記憶媒体の/d3/d4/にF1を格納する
    LUTに/d3/d4/F1/についての情報を書き込む
    GOTO END
}

```

ELSE {

K=1

LOOP2:

```

IF (K==M&&K==N) {GOTO ERROR}
ELSE IF (K==M&&K!=N) {K=K+1; GOTO LOOP2}
ELSE {

```

```

    IF (K番目の媒体にF1を格納する空き容量がある) {
        K番目の媒体にF1を格納する
        LUTに/d3/d4/F1について物理アドレス、セグメント

```

などの情報を書き換える

GOTO END

}

ELSE {

```

IF (K=N) { GOTO ERROR }
ELSE { K=K+1 GOTO LOOP2 }
}
}

```

ENDofLOOP2:

}

ERROR: 格納できない旨のメッセージを発行

END:

【0120】(3) ファイルの読出し
 ユーザの視点、若しくはデスクトップ上で、例えば/d5/d6/d7/d8というディレクトリに存在するファイルF2を読み出す操作を印加する。システム100の記憶空間内で、/d5/d6/d7/d8/F2が存在すれば、それを読み出す。なければエラー状態になり、そのパスにF2が存在しないというメッセージがユーザに通知される（例えば、デスクトップ上にダイアログが出現する）。

LUTを参照する

```

IF (LUTのエントリに/d5/d6/d7/d8/F2に関するエントリがある) {
    それが実際に格納されている記憶媒体の番号をLUT参照により知る
    (M番目の媒体だったとする)
    M番目の媒体から/d5/d6/d7/d8/F2を読み出す
    GOTO END
}

```

```

ELSE {
    GOTO ERROR
}

```

ERROR: ファイルが存在しない旨のメッセージを発行する

END:

【0123】(4) ファイルのコピー
 ユーザの視点、若しくはデスクトップ上で、例えば/d5/d6/d7/d8というディレクトリに存在するファイルF2を/d9というディレクトリにコピーする操作を印加する。システム100の記憶容量に、新たなファイル/d9/F2を格納するための空きがあれば、/d9/F2という複製を格納する。空き容量がなければ、エラー状態になり、空き容量がない旨のメッセージがユーザに通知される（例えば、デスクトップ上にダイアログが出現する）。

【0124】システム100の内部では、まず/d5/d6/d7/d8/F2を読み出す。そのときの手順は、上記の『(3) ファイルの読出し』で説明した通りである。

【0125】このファイル読出しに成功したら、次いで、/d9/F2という新規ファイルの格納を行う。その手順は、上記の『(1) 新規ファイルの格納』で説明

LUTを参照する

```

IF (/d10/d11/F3のエントリが存在した) {
    LUTにおける/d10/d11/F3に関するエントリを削除する
    LUTにおける/d10/d11/F3が格納されている記憶媒体に関する空き容量情

```

【0121】システム100の記憶空間からファイルを読み出すためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は(“*”は数字)、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0122】

【数7】

した通りである。

【0126】(5) ファイルの削除
 ユーザの視点、若しくはデスクトップ上で、例えば/d10/d11/F3と言うファイルを削除しようとする。そのファイルがシステム100の記憶空間に存在すれば削除する。なければ、そのようなファイルは存在しない旨のメッセージをユーザに返す（例えば、デスクトップ上にダイアログが出現する）。

【0127】システム100の記憶空間からファイルを削除するためのシステム内部の動きを、擬似コード形式で以下に示す。但し、“LOOP*”と“ENDofLOOP*”で挟まれる部分は(“*”は数字)、“goto”文により抜け出さない限り、繰り返し実行されるループ処理である。また、“IF”で始まる文は、条件判断文であり、それが満たされれば“{}”で囲まれた記述内容に従った動作を実行する。

【0128】

【数8】

報などを更新する

GOTO END

}

ELSE {

GOTO ERROR

}

ERROR: 削除すべきファイルがないというメッセージを発行

END:

【0129】(6) ファイルの移動

ユーザの視点、若しくはデスクトップ上で、例えば /d5/d6/d7/d8/F2 というファイルを他のディレクトリ、例えば /d12/d13/ に移動する操作を印加する。移動のための作業空間として必要な容量がシステム100に充分空いていなかったらエラーになる。

【0130】システム100内部では、まずLUTを参照して、/d5/d6/d7/d8/F2に関するエントリを探索し、これを削除する。

【0131】LUT上で、新たに /d12/d13/F2 というエントリを作り、/d5/d6/d7/d8/F2 に関して書いてあった記憶媒体番号、物理アドレスなどの情報を /d12/d13/F2 に関して書き込む。

【0132】[追補]以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。

【0133】本明細書では、コンピュータに対する適用例に基づいて本発明について説明してきたが、本発明の要旨はこれに限定されるものではない。例えば、ハード・ディスクやメモリなどの読み書き可能な記憶媒体を使用したファイル格納機能を持つ他のタイプの機器においても、同様にファイル分類の簡潔化という効果をもたらすことができる。

【0134】要するに、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【0135】

【発明の効果】以上詳記したように、本発明によれば、ハード・ディスクなどの記憶媒体を追加可能に構成されたコンピュータ上で、ディレクトリ構造を用いて多数のファイルを好適に管理することができる、優れたファイル管理装置及びファイル管理方法を提供することができる。

【0136】また、本発明によれば、ユーザから見たディレクトリの本構造を変化させずに記憶媒体を追加して記憶容量を増加させることができる、優れたファイル管理装置及びファイル管理方法を提供することができる。

【0137】本発明に係るファイル管理装置及びファイル管理方法によれば、ハード・ディスクやメモリなど読み書き可能な記憶媒体を使用するファイル格納機能におけるファイル分類を簡潔にすることができる。

【0138】また、本発明に係るファイル管理装置及びファイル管理方法によれば、コンピュータなどの情報処理システムにおいてハード・ディスクを追加したときに、既存のディレクトリによるファイル分類の形態をそのまま踏襲して、記憶容量が増えたという使い勝手をユーザに提供することができる。

【図面の簡単な説明】

【図1】記憶媒体を増設して新しいファイルを保存するときに実行するファイル管理の手順を模式的に示した図である。

【図2】編集作業などによってサイズが増大して、元の記憶媒体に保存できなくなったファイルを別の記憶媒体上に保存するときに実行するファイル管理の手順を模式的に示した図である。

【図3】元の記憶媒体に作成できなくなったディレクトリを別の記憶媒体上に作成するときに実行するファイル管理の手順を模式的に示した図である。

【図4】複数の記憶媒体にまたがってファイル参照するときに実行するファイル管理の手順を模式的に示した図である。

【図5】本発明に従ってファイル管理を行った場合に、ユーザに映るディレクトリ構造を模式的に示した図である。

【図6】本発明を実現可能なコンピュータ・システムのハードウェア構成を模式的に示した図である。

【符号の説明】

100…コンピュータ・システム

101…CPU, 102…メモリ

103…ディスプレイ・コントローラ

104…ユーザ入力インターフェース

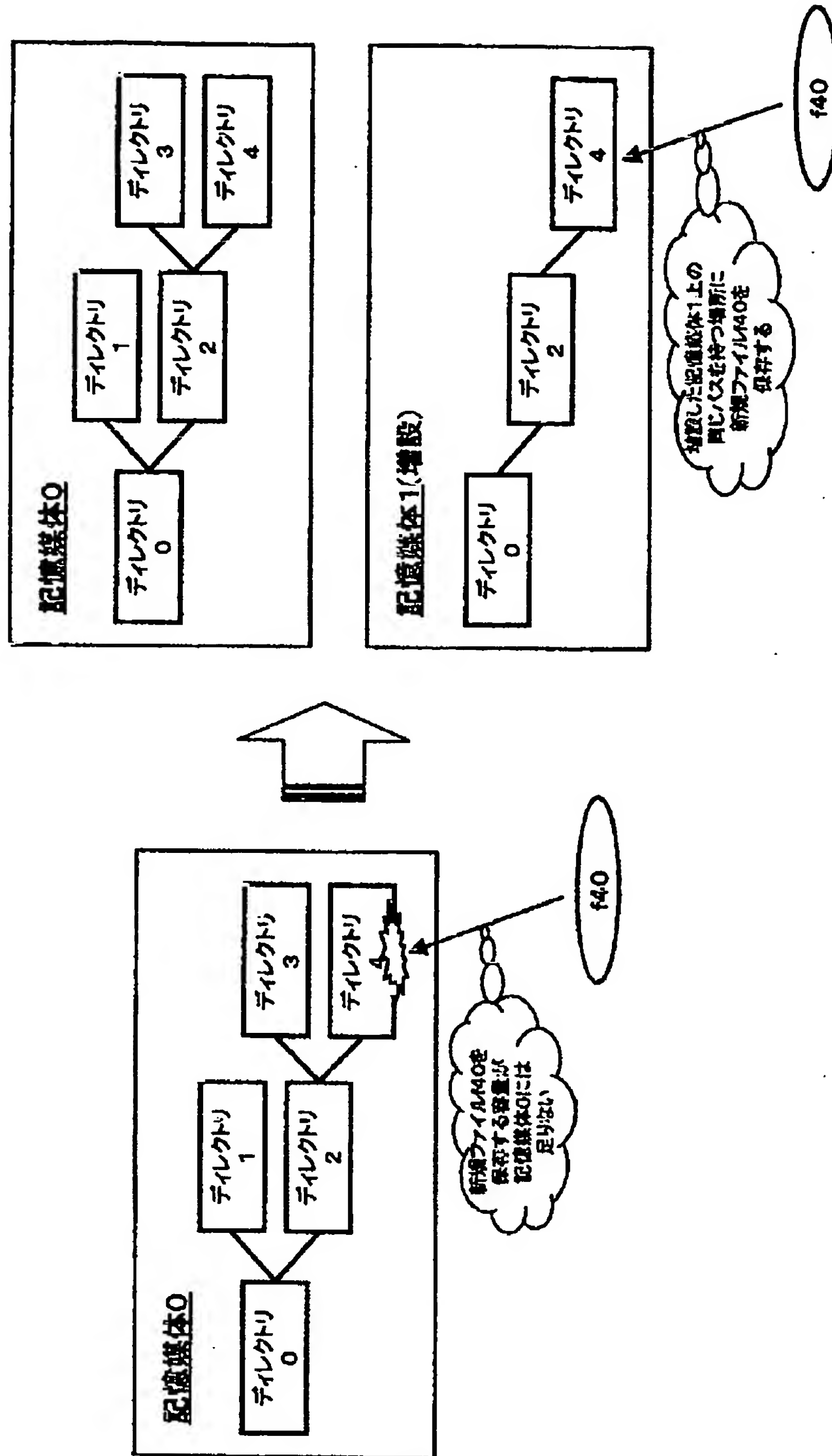
105…ネットワーク・インターフェース

107…外部機器インターフェース, 108…バス

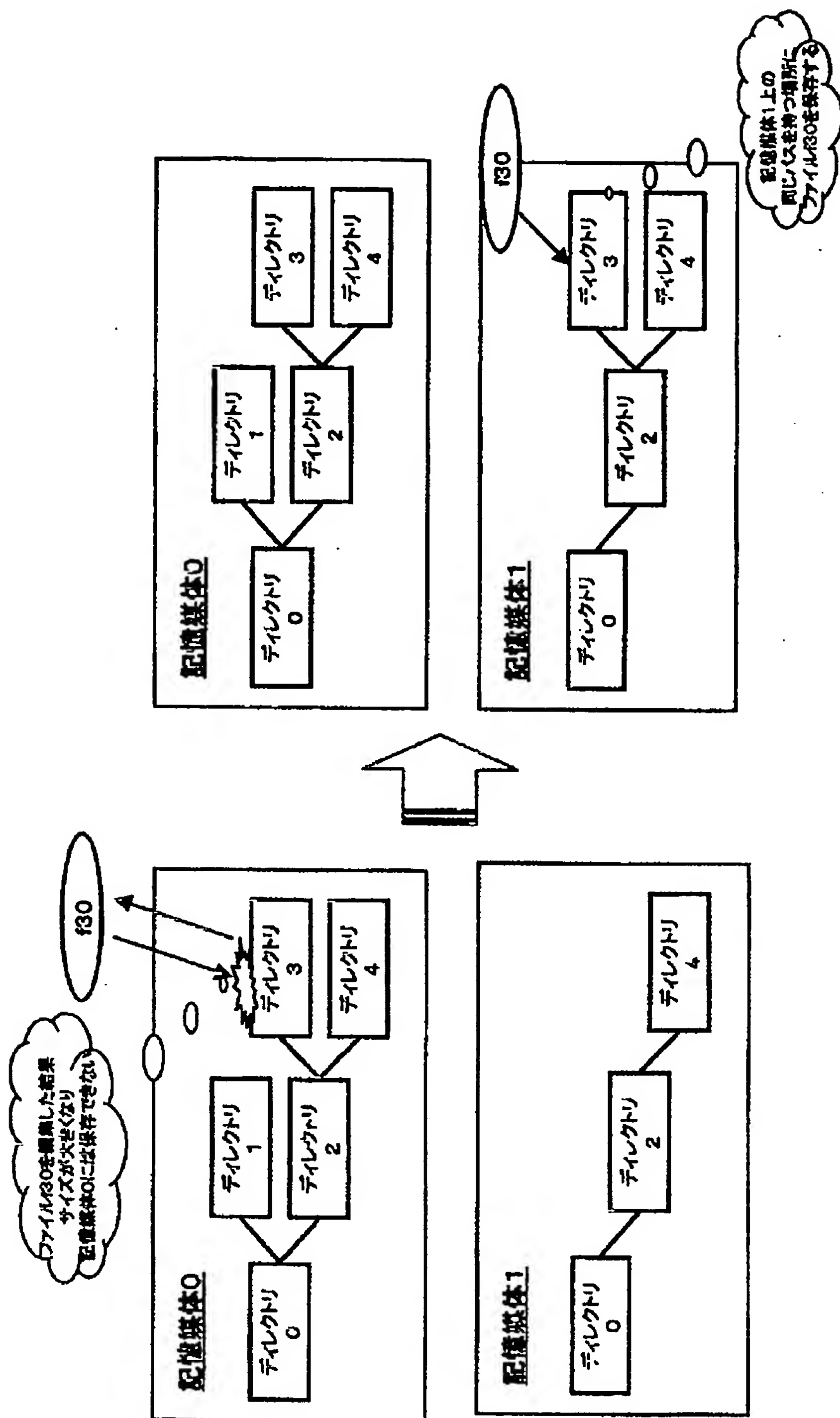
111…ディスプレイ, 112…キーボード, 113…マウス

150…記憶媒体

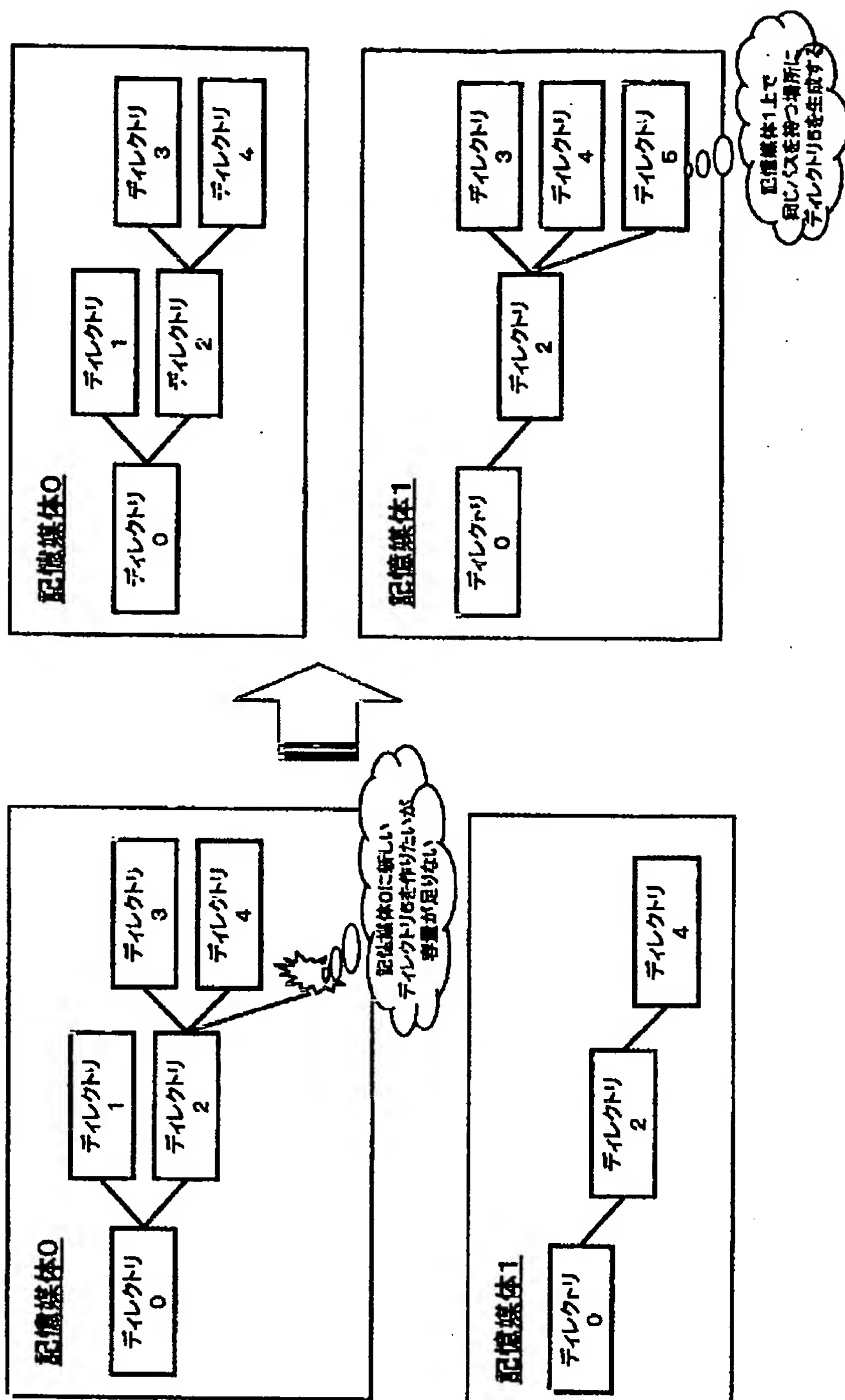
【図1】



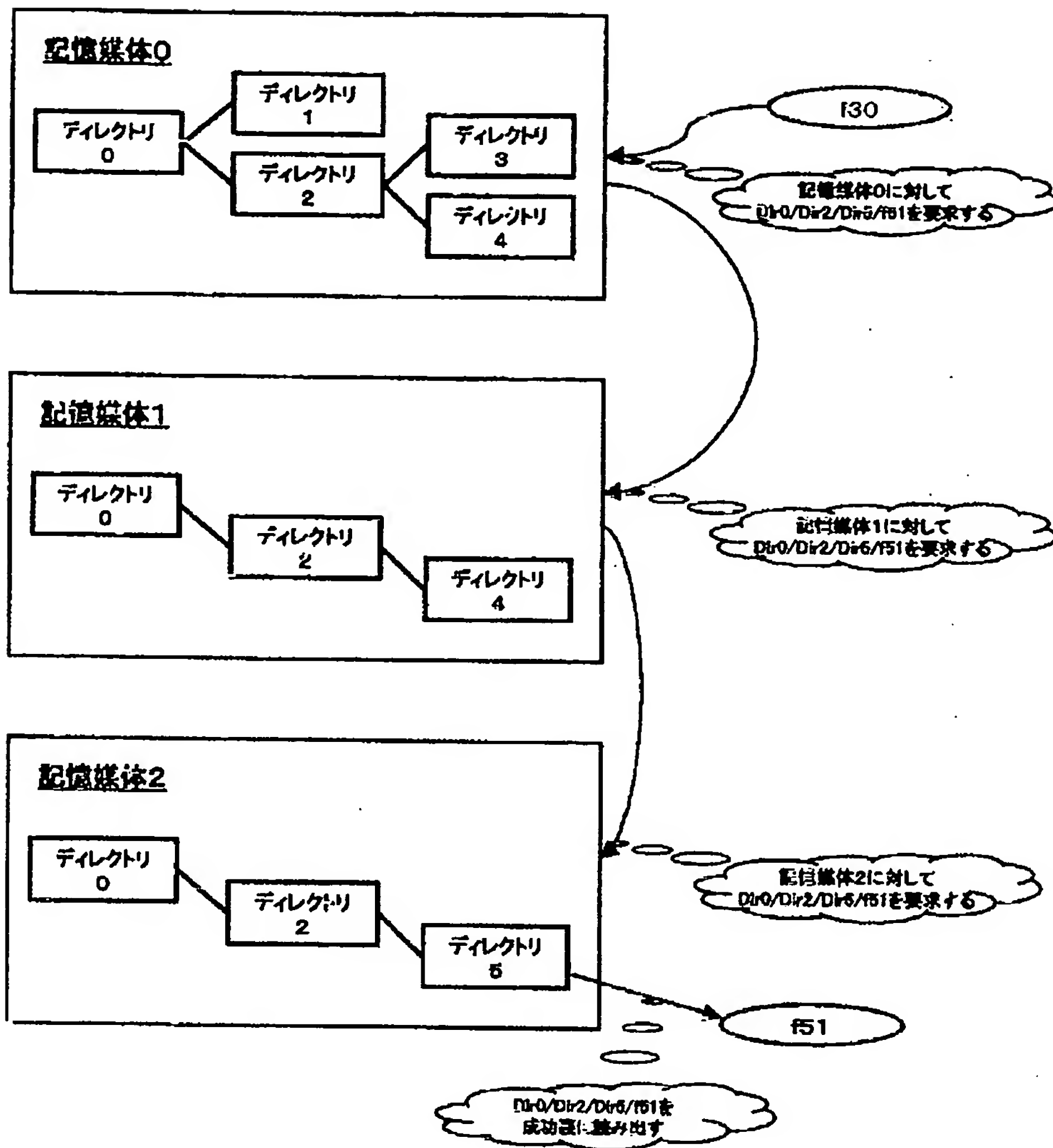
【図2】



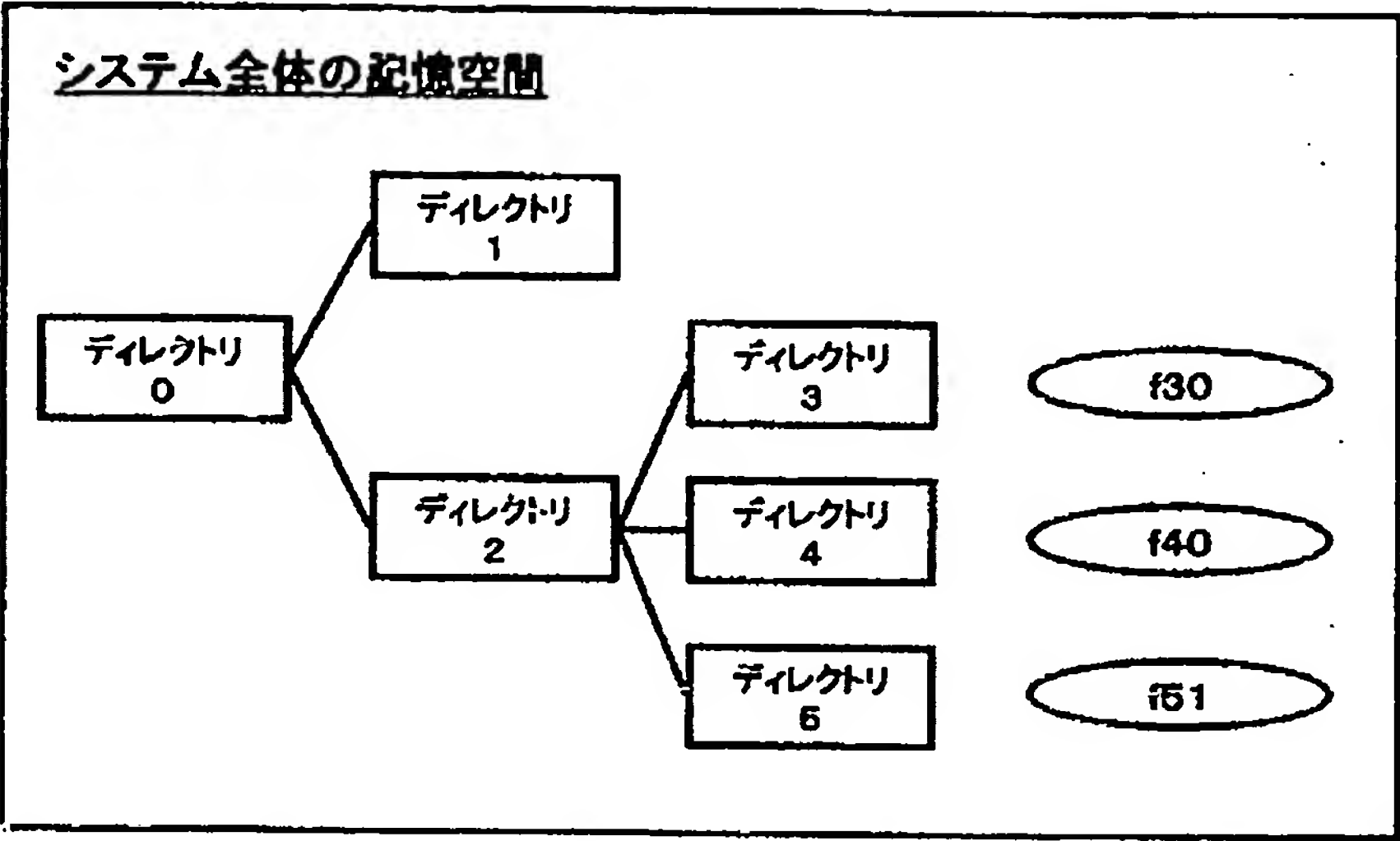
【図3】



【図4】



【図5】



【図6】

